

Reconocimiento óptico de caracteres mediante imágenes en contadores de gas

**Manuel Somoza Barreiro
Lidia María Jiménez Gutiérrez**

Madrid, junio de 2016



**Trabajo Fin de Grado del Grado en Ingeniería del Software
Departamento de Ingeniería del Software e Inteligencia Artificial
Facultad de Informática
Universidad Complutense
Curso 2015-2016**

Director: Gonzalo Pajares Martinsanz



“Me interesa el futuro porque es el sitio donde voy a pasar el resto de mi vida.”

Woody Allen



Autorización de difusión y utilización

Los abajo firmantes, matriculados en el Grado de Ingeniería del Software impartido por la Facultad de Informática, autorizan a la Universidad Complutense de Madrid (UCM) a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a sus autores el presente Trabajo de Fin de Grado, ***Reconocimiento óptico de caracteres mediante imágenes en contadores de gas***, realizado durante el curso académico 2015-2016 bajo la dirección de Gonzalo Pajares Martinsanz en el Departamento de Ingeniería del Software e Inteligencia Artificial.

Así mismo autorizan a la Biblioteca de la UCM a depositarlo en el Archivo Institucional E-Prints Complutense con el objeto de incrementar la difusión, uso e impacto del trabajo en Internet, y garantizar su preservación y acceso a largo plazo.

Madrid, Junio de 2016

Manuel Somoza Barreiro

Lidia María Jiménez Gutiérrez

Tutor: Gonzalo Pajares Martinsanz



Agradecimientos Personales

Desde aquí queremos dirigir unas palabras de gratitud.

Manuel Somoza Barreiro

Me gustaría agradecer a todos los que me han acompañado durante esta etapa y han hecho posible que pudiera alcanzar mis metas. Parte de todos mis éxitos futuros os pertenecen.

A mi familia, por vuestro apoyo y paciencia. Siempre me habéis dejado tomar mis propias decisiones y sin vosotros esto no hubiera sido posible.

A mis amigos, por haber estado siempre ahí, animando y regalando buenos consejos. La mejor compañía tanto en la biblioteca como en las cervezas de después.

A Dani y Gon, por enseñarme que la universidad es mucho más que trabajos y exámenes. De vosotros me llevo los mejores recuerdos.

A mis compañeros de clase, prácticas y pasillo. Por todas las horas invertidas en formarnos y divertirnos.

A mi compañera Lidia, por habernos entendido tan bien durante este último año. Ha sido un placer hacer este proyecto contigo.

Por último un recuerdo especial para quien estuvo ahí desde el principio, me apoyó más que nadie y confió siempre en mí.

Gracias a todos.



Lidia María Jiménez

Me gustaría agradecer en especial a mis padres por el esfuerzo que han hecho para que mi hermana y yo lleguemos donde queramos llegar y seamos lo que somos, en especial a mi madre, sin ella y su tesón, fuerza y confianza seguramente no estaría escribiendo hoy estas palabras. Gracias Mamá.

A mis abuelos Joaquín y Enriqueta y mis tíos Joaquín y Alexandra, gracias por su apoyo y consejos.

A mi hermana, por ser quien ve el mundo tan diferente a mí, que me hace cambiar la perspectiva de la vida, Gracias.

A Pedro Santamaría sin ti y el trabajo en equipo elaborado durante estos años acompañados siempre de mi madre, ser quien soy y esta carrera de fondo que llega a su fin no hubiera sido posible, gracias por todo.

A mis amigas y amigos, por estar siempre a mi lado, por darme el aire cuando lo necesitaba y la fuerza en mis momentos más bajos, nunca tendré las suficientes palabras de agradecimiento hacia vosotros.

Y a mi compañero del proyecto y prácticas, me alegro que nos hayamos entendido desde el primero momento siendo gratificante trabajar juntos.

A todos los que en algún momento me habéis acompañado en este largo camino haciéndome ser mejor, GRACIAS.



Agradecimientos

Agradecimientos a la Universidad Complutense y Madrileña Red de gas.

Ante todo agradecer al Departamento de Inteligencia Artificial e Ingeniería del Software de la Facultad de Informática por confiar en nosotros y poner a nuestra disposición toda la ayuda necesaria para llegar a este punto de nuestra carrera. Mención especial para nuestro director del Trabajo Fin de Grado Dr. Gonzalo Pajares por enseñarnos durante nuestra carrera en varias asignaturas y ahora en el TFG lo que es la inteligencia artificial y aportarnos todos sus conocimientos, por ayudarnos y estar siempre a nuestra disposición además de aportarnos la confianza necesaria en el momento oportuno.

Gracias



Contenido

Autorización de difusión y utilización	3
Agradecimientos Personales	4
Agradecimientos	6
TABLA DE FIGURAS	9
Lista de Abreviaturas	11
Resumen	12
Abstract	13
CAPÍTULO 1: INTRODUCCIÓN	15
INTRODUCTION	17
1.1 Objetivos y Organización del trabajo	19
1.2 Estructura de la memoria	20
1.3 Contribuciones Personales.	22
1.4 Decisiones de Desarrollo	26
CAPÍTULO 2: CONCEPTOS TÉCNICOS BÁSICOS	29
2.1 Visión por Computador	29
2.2 Tratamiento y procesamiento digital de imágenes	30
2.3 Aprendizaje Automático	30
2.4 Reconocimiento Óptico de Caracteres (OCR)	32
CAPÍTULO 3: TECNICAS Y METODOLOGIAS DE TRATAMIENTO DE IMÁGENES ..	35
3.1 Análisis de Imágenes	35
3.1.1 Imagen de Intensidad	36
3.1.2 Métodos del valor umbral	36
3.1.3 Apertura Morfológica (Opening)	38
CAPÍTULO 4: ANALISIS Y DISEÑO	39
4.1 Especificación Casos de Uso	39
4.1.1 Caso de Uso principal: Procesar imagen	43
4.1.2 Caso de Uso principal: Histórico de cliente	45



4.1.3 Caso de Uso principal: Listado de clientes	47
4.2 Diseño: Modelo–Vista–Controlador.....	48
4.2.1 Funcionamiento MVC:	50
4.3 MVC en el proyecto	50
4.3.1 Asociación Modelo–Vista–Controlador	51
4.3.2 Estructura e implementación en MVC.....	51
Modelo	51
Vista	53
Controlador	54
4.4 Arquitectura Software.....	54
CAPÍTULO 5: IMPLEMENTACION	57
5.1 Tecnologías utilizadas.....	57
5.2 Proceso de reconocimiento de lecturas	58
5.2.1 Recorte del área seleccionada.....	59
5.2.2 Reconocimiento de los dígitos.....	61
CAPÍTULO 6: CONCLUSIONES Y FUTURAS MEJORAS	63
CONCLUSIONS AND FUTURE WORKS	65
CAPÍTULO 7: BIBLIOGRAFÍA.....	67
CAPÍTULO 8: ANEXO	71
A) Manual de Instalación	71
B) Despliegue de la aplicación.....	73
C) Manual de Uso aplicación de escritorio.....	73



TABLA DE FIGURAS

Figura 1: *Relación de la visión por computador.*

Figura 2: Procedimiento para aprender un modelo de relación de datos.

Figura 3: Reconocimiento de Caracteres imagen a tratar (OCR).

Figura 4: Resultado de aplicar OCR a una imagen.

Figura 5: Tratamiento de imagen de intensidad sobre imagen del contador.

Figura 6: Tratamiento de imagen de umbralización.

Figura 7: *Tratamiento de imagen apertura morfológica sobre imagen del contador.*

Figura 8: Diagrama de flujo para el caso principal tratamiento de la imagen.

Figura 9: Diagrama de flujo para el caso principal Histórico del cliente.

Figura 10: Diagrama de flujo para el caso principal listado de cliente.

Figura 11: Diagrama de flujo del Caso de uso principal de tratamiento de imágenes desglosado por subcasos de uso.

Figura 12: Diagrama de flujo Histórico de cliente desglosado por subcasos.

Figura 13: Desglose del caso de uso principal listar Cliente por subcasos de uso.

Figura 14: Estructuración Modelo-Vista-Controlador.

Figura 15: Enfoque de la implementación MVC.

Figura 16: Primer encuadre del contador.

Figura 17: Selección del área de medición del contador.

Figura 18: Selección del área de identificación del cliente.

Figura 19: Área final recortada para el tratamiento de imágenes.



Figura 20: Imagen con el tratamiento de intensidad.

Figura 21: Resultado de aplicar el tratamiento de Binarización.

Figura 22: Resultado de aplicar el tratamiento de Apertura morfológica.

Figura 23: Consola de ejecución instalación MongoDB.

Figura 24: Captura de pantalla del menú principal de la aplicación.

Figura 25: Captura de la pantalla principal para el caso de tratamiento de imágenes.

Figura 26: Seleccionar imagen para cargar en el programa.

Figura 27: Carga de la imagen en el programa.

Figura 28: Selección del área de recorte.

Figura 29: Área recorta de un área previamente seleccionada.

Figura 30: Selección del área donde se encuentra los datos del cliente.

Figura 31: Selección del área donde se encuentra los datos del contador.

Figura 32: Confirmación del guardado de los datos.

Figura 33: Datos guardados correctamente.

Figura 34: Ventana del histórico de un cliente de la aplicación.

Figura 35: Ventana del listado de todos los clientes de la aplicación.



Lista de Abreviaturas

<i>ANPR</i>	<i>Automatic number plate recognition</i>
<i>EAN</i>	<i>European Article Number</i>
<i>HOG</i>	<i>Histogram of oriented gradients</i>
<i>IMR</i>	<i>Intelligent Machines Research Corporation</i>
<i>JS</i>	<i>JavaScript</i>
<i>LBP</i>	<i>Local Binary Patterns</i>
<i>MVC</i>	<i>Model-View-Controller</i>
<i>MVT</i>	<i>Model-Views-Template</i>
<i>OCR</i>	<i>Optical Character Recognition</i>
<i>PIL</i>	<i>Python Imaging Library</i>
<i>RGB</i>	<i>Red Green Blue</i>
<i>ROI</i>	<i>Región de Interés</i>
<i>SaaS</i>	<i>Software as a Service</i>
<i>SQL</i>	<i>Structured Query Language</i>
<i>SVM</i>	<i>Support Vector Machines</i>
<i>TFG</i>	<i>Trabajo Fin de Grado</i>
<i>UCM</i>	<i>Universidad Complutense de Madrid</i>
<i>CUPS</i>	<i>Código universal de Punto de Suministro</i>
<i>ROI</i>	<i>Region Of Interest</i>
<i>BD</i>	<i>Base de Datos</i>
<i>DVD</i>	<i>Disco Versatil Digital</i>



Resumen

La gran evolución a lo largo de este tiempo sobre dispositivos móviles y sus características, así como las vías de conexión de alta velocidad 3G/4G, han logrado dar un giro a los planteamientos económicos empresariales consiguiendo que se replanteen los costes de sus infraestructuras tradicionales, involucrando las nuevas tecnologías en su nueva estructura económica y consiguiendo invertir menos recursos humanos en el proceso de producción.

Este proyecto propone una solución real para la empresa Madrileña Red de Gas. Mientras el proyecto de contadores inteligentes se termina de concretar y desarrollar, es necesario disponer de un método que automatice la lectura de los contadores analógicos mediante el procesamiento de una imagen digital a través de una aplicación informática que sea capaz de determinar el código de identificación del contador así como la lectura del consumo actual.

Para la elaboración del método desarrollado se han utilizado conceptos propios de Visión por Computador y de Aprendizaje Automático, más específicamente tratamiento de imágenes y reconocimiento óptico de caracteres, mediante la aplicación de métodos en el ámbito de dichas disciplinas.

Palabras clave: Contadores de gas, Aprendizaje automático, Detección de Objetos, Visión por computador, Reconocimiento de caracteres, Percepción Computacional, tratamiento de la imagen.



Abstract

In the last time there has been a significant development in smartphones and their characteristics as well as the expansion of 3G/4G high-speed networks. These proposals change the direction of the companies' cheap plan, the evolution getting to raise again the infrastructure cost. Therefore they don't invest the same money that in traditional infrastructure. The companies use the new technology in the new structures' cheap, and then they use less human resources in the production.

This project is a real solution for the company Madrileña Red de Gas, while the intelligent gas meter project it's still on process. A method is needed to automate the analogs reading gas meter by applying Computer vision-based approaches and Machine Learning in order to identify the identification code in the gas meter and also the read for the current consumption. More specifically, image processing techniques and Optical Character Recognition are the methods applied for such purpose.

KEYWORDS: Gas meter, Machine learning, Character recognition, Perception computing, Computer vision, Object detection, Image processing.





CAPÍTULO 1: INTRODUCCIÓN

Madrileña Red de Gas es una empresa que opera en aproximadamente 59 Municipios de la Comunidad de Madrid contando con un total de 5.533 Kilómetros de distribución y dando soporte a compañías de su sector tales como Iberdrola, Endesa, Gas Natural Fenosa, etc.

La propuesta de proyecto viene motivada a raíz de la comunicación entre la empresa Madrileña Red de Gas y el director del presente trabajo hace ya un par de años, Madrileña Red de Gas describió el problema al que se enfrentaban con la llegada de las nuevas tecnologías informáticas y la automatización de la lectura de los contadores de gas. La empresa solicitó una aplicación informática capaz de llevar a cabo el proceso de lectura de los contadores de gas mediante el envío de una foto por el usuario al personal de gestión de la empresa vía procedimientos digitales, mientras la propia empresa, en paralelo a este proyecto, efectúa el cambio de contadores clásicos a contadores inteligentes, los cuales están dotados con la automatización suficiente de lectura y monitorizaran del consumo utilizando además aplicaciones o conexiones a la red.

El objetivo principal de este proyecto y de los gestores actuales de la empresa es ahorrar el desplazamiento de los sus empleados a los domicilios de los clientes para la lectura. De esta manera, a la vez que se automatizan los datos, se consigue minimizar el coste del procedimiento de lectura, reduciendo los riesgos de retraso en la instalación de los contadores inteligentes en los hogares de los clientes, que bien pudiera alargarse en el tiempo.

La primera toma de contacto con el proyecto se produce en el mes de Octubre de 2015 cuando a través del director del proyecto, el cual asume simultáneamente el rol de usuario final, se establecen los requisitos requeridos por la empresa.

En la actualidad, las lecturas de contadores son realizadas por los empleados cualificados de Madrileña Red de Gas, encargados de visitar cada domicilio a lo largo de los periodos de facturación, o bien mediante el envío por parte del cliente del dato relativo a la lectura del contador.

Esta forma de proceder tiene importantes inconvenientes, ya que es necesario que el cliente se encuentre en su domicilio para efectuar la medición del contador, o deja bajo su responsabilidad el envío de la información referente a su consumo del periodo establecido. Esto hace difícil la correcta recepción en tiempo y forma de la información a la empresa, aun habiendo buena voluntad de proceder por parte del cliente.



Parte de lo que se pretende resolver con este proyecto informático es la dependencia de la empresa a la transmisión manual de la información por parte de un empleado de la misma o un usuario del contador, que anotan la lectura y la envían por los procedimientos habituales, a través de la propia página web de la empresa o mediante correo electrónico.

Con este método, el usuario del contador será el encargado de tomar la medición del mes correspondiente mediante una fotografía, aumentando la fiabilidad así como la reducción de costes para la propia empresa. Al mismo tiempo, se reducen los casos en los que es necesario realizar estimaciones de consumo en base a los meses anteriores como sucede a día de hoy, método que puede dar errores en la facturación ya que el cliente puede tener un consumo no habitual, como por ejemplo en época de vacaciones o en momentos de exceso de demanda. Este problema se subsanaría en la siguiente lectura real del contador, pero las estimaciones pueden provocar desajustes que pueden ocasionar perjuicios económicos evitables.

La aplicación software que trata de resolver estas necesidades, está diseñada con el propósito de ser instalada en las máquinas disponibles en las oficinas de la empresa, de forma que la imagen del contador enviada por un cliente de la compañía pueda procesarse manualmente por un trabajador mediante el sistema de tratamiento de imágenes incorporado a la aplicación, cuyo objetivo es obtener la información correspondiente al identificador de usuario y la lectura de consumo. Ambos serán almacenados en la base de datos de la empresa para el próximo envío de facturación al cliente. Además, se dispondrá de un histórico del cliente donde se podrán consultar sus últimas mediciones.

Durante el curso 2014-15 se desarrolló una aplicación con similar finalidad [23], de tipo web y orientada al cliente de la compañía, donde el tratamiento de la imagen se realizaba de manera automática. No obstante, a pesar del aceptable funcionamiento de ésta, Madrileña Red de Gas optó por el nuevo diseño con el objetivo de aumentar la tasa de aciertos en el procesamiento de las imágenes recibidas. Para ello, se estudiaron nuevos métodos de procesamiento de imágenes, lo que derivó en un nuevo planteamiento de diseño conceptual y de implementación, y en un desarrollo integral de la aplicación.



INTRODUCTION

Madrileña Red de Gas is a business company that operates in 59 cities in Madrid with a total 5.533 km of network and it provides services to different companies in the sector, for example, Iberdrola, Endesa, Gas Natural Fenosa, among others.

The project proposal is the result of the communications between the Madrileña Red de Gas Company and our Project manager in the last years. The Company explained the problem to be solved with the new computational technologies in order to read the real automatic read in the gas meter. The computer-based application requested by the company calculates the real automatic read in the gas meter and the unique identification code. The customer sends an image to the Company by email. This process will be active during the transition period to change the classic gas meter to the new intelligent gas meter; the new gas meter will be automatic, monitoring the meter reading of the consumption based on intelligent approaches.

The main objective in the project of the analogue meters gas is to avoid the shift of the workers to go to the costumers addresses to read the gas meters, as well as, the automatic process for reading, This can minimize the cost of the process and this process will help to put the new intelligent counters of the house in progress on time.

In October 2015 we had our first contact with the project where we talked with the director. He takes the role as the final user and transmits the requirements from the company.

Currently this read is done for the specialist's employees. During the read process the employees travel to the customer home's in the billing term or the customer sends the relative piece of information by the gas meter gas. This traditional process involves a lot problems, for example, sometime the employees cannot get access into the costumers home's to read the gas meter because they are absent at this moment, in this case the costumer must send the information about the gas meter reading. Sometimes, the company receives incomplete and insufficient information despite of the costumer has positive wishes to send the correct information.

This project tries to improve and minimize the dependency of the Company from the gas meter manual readings by employees. With this process the costumer sends the information and the Company does not need to do the estimation based on past readings. This avoids problems with the billing when customer does not provide real readings. Gas meter Thus, to solve this problem



we design a new software-based tool, which is installed locally on computers available at the company. When the customer sends the image with the gas meter, the employees cut the area of interest and the image processing and machine learning methods embedded in application extract the measuring of gas meter together with the identification code. The employee insert this information in the Database specifically designed with such purpose for the next billing, as well as, to save historic records associated to each individual gas meter gas.

An application with similar objectives was developed during the last year [23], based on web services, where the images processing was fully automatic. Nevertheless, Madrileña Red de Gas put the emphasis on improving the rate of success during the image processing and character recognition. Under this new insight new image and vision computing approaches have been developed with the new conceptual application design, where some manual processes are involved, such as the selection of the area of interest in the image with the exclusive information to be extracted.



1.1 Objetivos y Organización del trabajo

La idea del proyecto surge de un objetivo real, basado en la necesidad de automatizar las lecturas de contadores por parte de la empresa Madrileña *Red de Gas*, cuya expansión cubre amplias zonas de gestión de lectura de contadores. Por lo tanto, el proyecto está basado en el desarrollo de software práctico para el ámbito empresarial cuya única finalidad es dar solución a un problema real en el ámbito productivo.

La primera toma de contacto con el desarrollo del proyecto fue una reunión con nuestro director en la cual nos expresó los deseos por parte de Madrileña Red de Gas. En esta primera reunión se realizó el planteamiento del problema y la definición de los requisitos, que consistían en la captura de imágenes para su posterior tratamiento con la finalidad de identificación de las cadenas de caracteres que constituyen tanto la lectura como el código de identificación del usuario.

Tras un estudio a fondo de la problemática a resolver, se planteó el desarrollo de una aplicación informática de escritorio en la cual un trabajador de la empresa puede cargar una imagen recibida, a la vez que captura y reconoce mediante recortes de imagen manuales, el identificador único de usuario y la información de consumo para el periodo de facturación correspondiente del cliente identificado.

Dado que el proyecto ya realizado mencionado anteriormente, estaba desarrollado en versión web, vimos durante el análisis que ningún material previo desarrollado era incorporable directamente al presente proyecto. Por ello, durante las primeras semanas se dedicó gran parte del tiempo y recursos a la investigación de tecnologías para el desarrollo que se adaptaran de forma óptima a las necesidades del cliente.

Tras esta primera fase de investigación y basándonos en los requisitos se llegó a la conclusión de desarrollar una aplicación en la que es el personal de Madrileña Red de Gas el encargado de subir la imagen del contador tomada en posición frontal y enviada por el cliente de la compañía por los procedimientos establecidos al efecto (web o correo electrónico), recortar las zonas donde se encuentra tanto la identificación del cliente como los datos del consumo del periodo sin las cifras decimales (por carecer de relevancia), obteniendo con estos recortes la información de consumo y el identificador de cliente correspondiente, que se almacenan en la Base de Datos de la empresa.



El objetivo principal es por tanto, conseguir una aplicación que consiga la menor tasa de fallos por procesamiento de imagen posible, intentando además conseguir un producto vistoso y amigable para los trabajadores de la empresa.

Los objetivos futuros planteados, son continuar el desarrollo de la aplicación para automatizar todo el proceso de tratamiento de imágenes manteniendo los niveles de acierto y la persistencia de la información en plataformas Cloud.

1.2 Estructura de la memoria

El presente documento se estructura en capítulos, de forma que en cada uno de ellos se analizan las partes del proyecto además de nuestra aportación al mismo, y los datos de interés para el análisis, así como los desarrollos llevados a cabo para completar la aplicación.

El presente capítulo que corresponde al primero, contiene la parte previamente descrita dedicada a la motivación y planteamiento inicial, que originan la aplicación, así como las contribuciones personales de cada componente del grupo junto con la ayuda y colaboración recibida, también se explican las decisiones preliminares tomadas para el desarrollo del proyecto, tanto a nivel de implementación como a nivel funcional del mismo.

En el segundo capítulo se realiza una recopilación de los conceptos básicos y de las técnicas algorítmicas utilizadas para el procesamiento de imágenes, Más específicamente, se establecen los planteamientos necesarios para aplicar técnicas de reconocimiento de imágenes en general y de caracteres en particular para el análisis de la partes de interés de las imágenes, esto es el reconocimiento del identificador de usuario y la lectura de consumo en la imagen conteniendo la parte frontal del contador.

En el tercer capítulo, se desarrollan y detallan las técnicas y metodologías utilizadas para el tratamiento de imágenes, en ellas se incluyen los métodos de tratamiento de imágenes para utilizar imágenes de intensidad, binarización, umbralización automática y apertura morfológica.

En el cuarto capítulo se desarrolla y describe la parte metodológica desde el punto de vista de ingeniería del software, donde se detallan aspectos tales como el patrón de implementación aplicado al proyecto, los casos de uso principales y cómo están subdivididos en subcasos de uso secundarios.

En el quinto capítulo se entra en detalle en las tecnologías utilizadas en el proyecto y en la implementación del mismo, incluyendo las partes relativas a



cómo se ha construido internamente tanto la parte visible para el usuario como la parte más interna del mismo, además de explicar el procedimiento del proceso de reconocimiento de la imagen.

En el sexto capítulo se exponen las principales conclusiones del proyecto tanto en inglés como en español, incluyendo las posibles mejoras futuras tanto desde el punto de vista del desarrollo como de la implementación.

En el séptimo capítulo se exponen las fuentes de información que se han ido consultando para el desarrollo del proyecto.

En el octavo capítulo contiene un anexo con los siguientes puntos:

- El punto A contiene toda la información necesaria para la instalación del software necesario y su utilización por parte del usuario final.
- El punto B contiene el despliegue del programa que es la información necesaria para la correcta puesta en funcionamiento de la aplicación.
- El punto C contiene un tutorial sobre el funcionamiento de la aplicación, donde el usuario va a encontrar imágenes explicativas de todas las funcionalidades del programa.



1.3 Contribuciones Personales.

Manuel Somoza Barreiro

Tras una primera reunión informal con el tutor en Mayo de 2015 para preseleccionar el proyecto, mi aportación al mismo comienza durante el verano realizando tutoriales online de Python, la herramienta principal requerida para el desarrollo de nuestra aplicación.

Una vez comenzado el curso, es en Octubre cuando se inicia la planificación del proyecto, después de una primera reunión oficial con el tutor para recoger la documentación del trabajo realizado hasta el momento por compañeros de cursos anteriores, establecer los requisitos tanto con el tutor, como con la empresa para la que se realiza la aplicación y establecer los primeros hitos y fechas de entrega.

Durante el siguiente mes, mi labor se centró en el **análisis de los requisitos** y la **obtención de los casos de uso** para después de un par de reuniones con mi compañera Lidia repartirnos las primeras tareas.

En ese reparto, mis labores fueron el **desarrollo del módulo de integración y el módulo de negocio**, responsable de todo el funcionamiento interno, dejando la parte de la interfaz para mi compañera.

Primero, para desarrollar el módulo de integración, comencé investigando que sistema de base de datos resultaba mejor en nuestro caso para la persistencia de la información.

Decidí utilizar MongoDB y su librería PyMongo para Python como reto formativo ya que no tenía experiencia en él y todo mi trabajo de desarrollo en este proyecto lo he enfocado en el aprendizaje y el uso de herramientas que no hubiera utilizado previamente.

Una vez desarrollado el módulo, realicé pruebas caja blanca para comprobar que los flujos de datos eran correctos y que la persistencia y la carga de datos se realizaban correctamente.

A continuación, me dediqué a investigar sobre tratamiento de imágenes y sistemas de reconocimiento de caracteres en entornos de desarrollo Python. Quizás esta fue la parte más laboriosa de mi participación en este proyecto ya que tuve que buscar librerías, ejemplos de uso y realizar diferentes pruebas aisladas para ver si las funcionalidades que me ofrecían eran las mejores para incluir en nuestra implementación.



Tras todo ese proceso, decidí utilizar tras recomendación del tutor OpenCv como librería para el tratamiento de las imágenes (abrir una imagen, mostrarla, convertir a escala de grises, binarización, apertura morfológica...) y Tesseract OCR para el reconocimiento de los caracteres.

Con estas librerías desarrolle el sistema de recorte manual de una imagen, todos los métodos para su tratamiento y su lectura a través de Tesseract para obtener la información de nuestras imágenes de contador.

Después de realizar diferentes pruebas para garantizar el buen funcionamiento e intentar mejorar la tasa de aciertos en el reconocimiento de los caracteres aplicando mejores filtros a la imagen, me ocupé del **desarrollo del controlador** para ir ensamblando los módulos.

En este punto, después de mostrar al tutor nuestros avances, y tener cerrada ya toda la funcionalidad de la aplicación, tras una nueva reunión con mi compañera decidimos volver a reasignar tareas.

Puesto que Lidia estaba encontrando dificultades en el desarrollo de la interfaz, decidimos que ella se encargaría de continuar con el desarrollo de la memoria y yo me haría cargo de finalizar la interfaz, ya que las destrezas adquiridas con el resto de la aplicación me podrían ayudar a finalizar este módulo.

Para ello, después de investigar que librería GUI nos podría ayudar a solventar nuestras dificultades, decidí continuar con Tkinter para el **desarrollo de la interfaz** y comencé un proceso de lectura de manuales y pruebas aisladas para acabar incorporando en nuestro desarrollo, no sin muchas dificultades, las funcionalidades para acabar teniendo una aplicación de escritorio intuitiva y bien diseñada.

Durante todo el proceso de desarrollo, he ido realizando una **guía de instalación** de todos los módulos y librerías para poder portar la aplicación a otros equipos.

Como complemento al desarrollo de la aplicación, y a la realización de las guías, durante todo el proceso he tenido **reuniones con mi compañera** para ayudarle a definir cómo plasmar en la memoria el contenido referente al análisis y desarrollo de la aplicación.

Durante las semanas antes de la entrega definitiva, realicé diferentes **revisiones técnicas formales** de toda la memoria para solucionar posibles errores de contenido y forma que hubieran podido pasar desapercibidos, como fueron por ejemplo tener reescribir casos de uso o definición de funciones y clases. También realice ajustes en la maquetación previa a la entrega.



Lidia María Jiménez Gutiérrez

Mi colaboración en el proyecto comenzó con una reunión con nuestro director del trabajo en el mes de Octubre para recibir información detallada de los requisitos del proyecto, así como de cuáles eran los objetivos que se querían alcanzar a la finalización del proyecto en el mes de junio.

Una vez establecidos con claridad los objetivos y los tiempos disponibles para el desarrollo del proyecto, las dos primeras semanas las dediqué a la lectura de la documentación facilitada por el director, leí y estude la documentación enfocando mi atención en los puntos relacionados con la memoria, también me documenté sobre qué puntos obligatorios nos pedían en la memoria y además, cómo debía estar estructurada y el contenido que debía de tener como requisitos mínimos. El objetivo era tener una base de información suficiente de la que partir para estructurar el presente proyecto y la entrega final, con toda la información que recapitule en esas dos semanas nos reunimos y decidimos la forma de enfocar nuestro proyecto tras lo cual se concretó una nueva reunión con el director que supuso el arranque del mismo.

Estuvimos otras 3 semanas recopilando información sobre lenguajes de programación y análisis de imágenes que necesitaríamos para la implementación del software. En este punto, estuve buscando información sobre compiladores de Python, además de documentarme de cómo instalar el compilador. Sobre las clases de tratamiento de imágenes estuve buscando tanto en internet como en la documentación que guardaba de la asignatura *Percepción Computacional*, impartida por el propio director cuyo contenido, en un elevado porcentaje se basó en el tratamiento de imágenes.-El principal problema derivó del lenguaje de programación elegido ya que las clases que implementaban parte de nuestro software estaban en Python y las clases disponibles para la implementación del análisis de imágenes estaban en MATLAB, por lo tanto tuve que buscar para Python los tratamientos de imágenes a utilizar y determinar cómo podíamos integrarlas en el proyecto para que el producto final funcionara correctamente. Además de lo anterior realicé un curso de programación de Python para adaptarme al entorno de programación y tener claras nociones básicas del lenguaje.

Una vez disponibles todos los métodos software necesarios para la implementación, las mismas versiones y todas las clases a utilizar, procedimos a la distribución eficiente del trabajo entre los componentes del equipo. Dado que mi compañero ya tenía funcionando la base de datos, decidimos de mutuo acuerdo que él se iba a encargar de la parte interna entendiendo como tal la parte no visible para el usuario y yo en una primera instancia de la parte externa (visible al usuario), en palabras técnicas mi función dentro de la



implementación del software iba a ser el desarrollo de la interfaz gráfica del programa.

Para la elaboración de la interfaz lo primero que tuve que hacer es documentarme sobre clases que ofreciera Python para su desarrollo, al final me decante por Tkinter ya que era la más simple y completa, realicé con ella una primera versión básica de la interfaz para poder comprobar que el trabajo de mi compañero enlazado con el mío funcionara, después de esta primera parte, tuve que elaborar mejoras, la primera mejora fue la imagen estética de la interfaz para ello quise poner fondos de imágenes que aparecían en la propia página web de la empresa y tuve mi primer problema, tkinter [24] no dejaba ponerlas de fondo, estuve documentándome y al final me decanté por la clase PhotoImage[15], con ella pude insertarlas y hacer una segunda versión de la interfaz. Probando la interfaz con la parte de mi compañero nos dimos cuenta que la interfaz no era válida ya que si se abría la interfaz en otro tamaño de pantalla, se descolocaba, por lo que la siguiente mejora de la interfaz era introducir “frames” en la interfaz para que fuera compatible con cualquier tamaño de pantalla, en este punto fue cuando me encontré con el mayor de los problemas ya que si creaban los “frames”, el controlador no interactuaba, fue en este punto cuando mi compañero con más experiencia en Python decidió seguir con ello y yo decidí volcarme en la elaboración memoria del proyecto.

La memoria ha sido elaborada por mí en su totalidad incluyendo las traducciones a la lengua extranjera obligatoria como parte de la documentación y excluyendo los puntos que tenían que ver con las explicaciones personales y obligatorias de mi compañero y el apéndice de la instalación del software necesario. La elaboración de la memoria constituyo a la vez un elemento determinante del progreso y avance del proyecto, así como del análisis y valoración de los resultados obtenidos, tanto desde el punto de vista operativo de la aplicación como desde el punto más específico relativo a determinar si los métodos de identificación del código de usuario y lectura de contador eran lo suficientemente eficaces. En este sentido y bajo este punto de vista, se puede afirmar, que asumí el rol correspondiente a gestión del proyecto, tal y como se contempla desde la perspectiva de la ingeniería del software.

Para finalizar el proyecto, durante los meses de Abril y Mayo nos reunimos con mayor intensidad en sucesivas sesiones con nuestro director para realizar un seguimiento de los avances y futuros pasos a tomar y de forma individual, durante la última semana de Mayo mi compañero y yo nos hemos reunido de forma continuada para revisar todos los planes del proyecto, junto con la realización y verificación de las pruebas finales y finalmente mi aportación termino con la maqueta y finalización de la memoria, incluyendo las correcciones de la misma.

1.4 Decisiones de Desarrollo

Durante las primeras fases de ejecución del proyecto se definieron los requisitos tanto de diseño como de desarrollo demandados por la empresa Madrileña Red de Gas.

Las decisiones de desarrollo se dividen en dos bloques diferenciados, por un lado los llamados requisitos funcionales del software donde se contemplan las acciones y características que debe cumplir la aplicación desarrollada al final del proyecto y por otro lado un segundo bloque conteniendo los requisitos técnicos del software, que son todos aquellos relacionados con el código, lenguaje de programación, compiladores o compatibilidad de versiones. A continuación se detallan ambos con mayor profundidad.

Los requisitos funcionales son los relacionados con la lógica de funcionamiento, estos requisitos fueron sencillos de definir ya que la empresa estableció con claridad los conceptos a tener en cuenta, los acuerdos son los siguientes:

- El software debe poder ser instalado en los dispositivos de la empresa.
- El usuario debe ser un trabajador de la propia empresa.
- La recepción de la imagen se realiza por vía digital, mediante correo electrónico o desde la propia página web de la empresa.
- Existirá una Base de Datos como almacén de información.
- Interfaz sencillo y amigable para el trabajador, ya que estará una parte importante de su jornada laboral trabajando con la aplicación.
- El sistema operativo debe ser Windows ya que es el sistema instalado en los ordenadores de la empresa, con posibilidad de migrarlo a otras plataformas en un futuro.
- En caso de que el programa interprete mal la lectura del contador o la identificación del cliente, el usuario debe poder corregirlo manualmente e insertarlo en la base de datos.
- Recortar la imagen eligiendo sólo la región limitada por el código de lectura y el código de identificación de cliente, lo que constituyen las regiones de interés desde el punto de vista del procesamiento de la imagen.
- Los recortes de las regiones de interés deben efectuarse manualmente ya que se mejora la tasa de acierto aunque se pierda usabilidad.
- El algoritmo desarrollado debe ser capaz de realizar el reconocimiento de caracteres con independencia de las condiciones ambientales donde esté ubicado el contador, que afectan principalmente a la luminosidad de la imagen a tratar.



Las decisiones tomadas para cumplir los requisitos funcionales están desarrolladas y explicadas en los casos de uso, apartado 4.1.

En relación a los requisitos técnicos, éstos hacen referencia a las características que debe cumplir el hardware del equipo donde va a estar instalado el programa. Ello deriva en la necesidad de desarrollar un software compatible con esas características para su correcta instalación y utilización. En esta categoría de requisitos se incluyen los relacionados con la implementación del software, que estaría condicionado por las características del código, programación, instalación, compatibilidades y librerías necesarias para la creación y la determinación de la funcionalidad del programa a desarrollar. Los requisitos técnicos establecidos al comienzo del proyecto son:

- Lenguaje de programación a utilizar debe ser Python, requisito del director del proyecto para dar continuidad, en la medida de lo posible a desarrollos de cursos previos.
- Lenguaje de programación compatible con las clases de tratamiento de imágenes existentes.
- En aras de la reusabilidad del software, se debe procurar la utilización de clases ya existentes, con funcionalidades desarrolladas para el tratamiento de imágenes con el fin de evitar desarrollos innecesarios.
- Utilización en la medida de lo posible de Software Libre, código abierto o gratuito.
- Compatibilidad del sistema operativo Windows con la base de datos elegida.

A partir de estos requisitos técnicos se decidió que la mejor librería para el tratamiento de imágenes existente era OpenCV [5] la cual proporcionaba la mayor parte de las funciones necesarias y previstas en los requisitos de la implementación relativas al tratamiento de imágenes, siendo además gratuita y compatible con dos lenguajes sencillos e intuitivos. No obstante, estuvimos sopesando la posibilidad de utilizar MATLAB [9] ya que lo habíamos usado con anterioridad en la asignatura de *Percepción computacional* y algunos de los algoritmos que íbamos a necesitar ya estaban totalmente disponibles y operativos. Sobre el lenguaje de programación debatimos entre C++, que habíamos usado con anterioridad en la asignatura de *Estructura de datos y algoritmos y Técnicas algorítmicas en Ingeniería del Software* con el que ya estábamos familiarizados con el entorno de compilación o Python [1] que proporcionaba mucha más información sobre la librería Open [5]. Considerados los pros y los contras se decidió finalmente una implementación en Python.



Otra de las decisiones que tuvimos que afrontar fue la base de datos a elegir, que fuera compatible con el lenguaje de programación propuesto por el director y que generara el menor conflicto posible con las librerías OpenCV[5]. Al decidir desarrollar un software para escritorio la base de datos que mejor se adaptaba era DB MONGO [13], una de las más utilizadas del tipo NoSQL [34] y orientada a documentos. Lo que significa que en vez de guardar los datos en registros, lo hace en documentos que son almacenados en BSON [35], que es a su vez una representación Binaria de JSON [33]. Como elemento novedoso que incluyó una parte de aprendizaje desde nuestra formación, nos encontramos con la novedad de que en Mongo no existen tablas, ni registros, ni SQL, lo cual resultaba ciertamente atractivo.

Finalmente nos quedaba tomar la decisión en la parte relativa al reconocimiento de imágenes, más exactamente con orientación a la identificación de una secuencia o serie de números para poder identificar el código del contador y la lectura de consumo. Tras el correspondiente análisis de posibilidades se optó por OCR [3], en gran medida esta decisión fue debida al hecho de que ya se había utilizado en el trabajo desarrollado con anterioridad [23]. En este sentido, lo primero a realizar era obtener el dato contenido en el contador que constituye lo que se denomina ROI (Region Of Interest), de forma que una vez obtenido dicho dato puede convertirse mediante OCR a una serie numérica. Con el fin de poder seleccionar correctamente la ROI en la imagen, evitando que la parte de interés no se mezcle con componentes de color rojo que aparecen como elementos externos carentes de interés, como por ejemplo tuberías del gas, partes de la pared, reflejos de luz, se ha utilizado e implementado la opción de recorte de imagen a partir de la librería Open CV mencionada con anterioridad, en el propio código ha sido llamada a esta funcionalidad de OpenCV a través de la clase cortar imagen explicada en el apartado 4.3.

CAPÍTULO 2: CONCEPTOS TÉCNICOS BÁSICOS

En este capítulo se abordan los conceptos técnicos investigados y desarrollados para la realización del proyecto, proporcionando un enfoque general sobre la evolución de la visión por computador, el tratamiento de imágenes, el aprendizaje automático y la percepción computacional en el ámbito de la lectura de caracteres y la búsqueda de regiones dentro de una imagen.

2.1 Visión por Computador

La Visión por Computador también conocida como Visión Artificial o Visión de Máquina, está incluida en el ámbito de la Inteligencia Artificial, incluyendo una serie de métodos encaminados a procesar, analizar y llegar finalmente a la comprensión del contenido de las imágenes. El fin principal es conseguir que un computador sea capaz de recibir una información de la escena tridimensional, en forma de imagen, a través de sensores (cámaras que capturan la imagen, bien estáticas o mediante secuencias de video). En esas imágenes es donde está contenida la información numérica o simbólica que se utiliza en el presente proyecto. Lo que se pretende es llegar a una aproximación lo más real posible de la visión humana, incluyendo habilidades y capacidades.

En la figura 1 se muestra un esquema de las relaciones existentes entre la Visión y otras disciplinas en las que se basa y sobre las que se aplica.

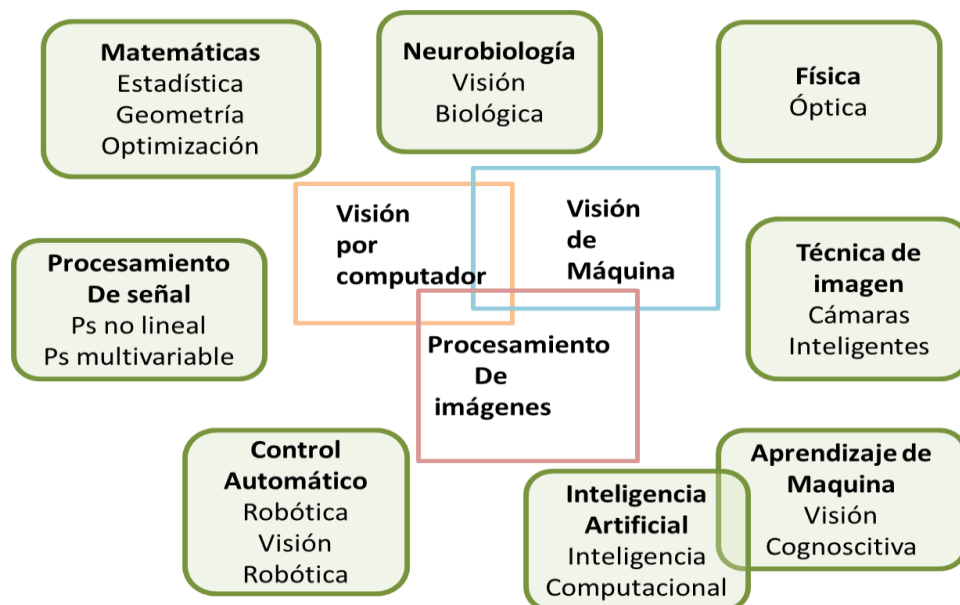


Figura 1: Relación de la visión por computador.



En la actualidad existen diversas áreas de aplicación, como indica la figura 1, destacando algunas como la medicina, robótica o seguridad, dentro de cada una de ellas se procesan distintas disciplinas.

2.2 Tratamiento y procesamiento digital de imágenes

El procesamiento digital [14] de imágenes se refiere a la aplicación de un conjunto de técnicas cuyo objetivo consiste en transformar una imagen para obtener otra imagen resultante, la cual contiene información más explícita de la escena que la original, además de mejor calidad, de forma que desde nuestro punto de vista facilitará el reconocimiento de los códigos que constituyen el objeto de interés de las lecturas de consumo e identificación de los clientes.

Entre las técnicas más conocidas en el tratamiento y procesado digital de imágenes se puede mencionar el proceso de filtrado en sentido amplio cuyos objetivos se encaminan a suavizar la imagen, eliminar ruido, realzar bordes. Desde el punto de vista del tratamiento de señales, como lo es una imagen, el filtrado en el dominio de la frecuencia, utilizando la transformada de Fourier [30] es una técnica ampliamente utilizada para mejorar la calidad de la imagen o extraer información relevante como bordes u otras estructuras significativas. El filtrado en el dominio espacial [31] tiene la misma finalidad, si bien en este caso se lleva a cabo directamente sobre los píxeles de la imagen mediante técnicas de convolución.

Además de las técnicas de filtrado antes mencionadas, existen otras muchas para el procesado de las imágenes de interés desde el punto de vista de la aplicación desarrollada, a saber: detección de bordes, reconocimiento de líneas, análisis de histogramas, mejoras del contraste, eliminación del ruido, segmentación de regiones.

2.3 Aprendizaje Automático

Bajo el concepto de aprendizaje máquina se engloban un conjunto de técnicas y algoritmos para extraer información de unos datos o bien para estimar una dependencia o estructura desconocida de un sistema, utilizando un número limitado de observaciones de entrada-salida [7].

La extracción de información de los datos está en la base misma de la investigación científica. Las personas acumulamos información sobre nuestro entorno, organizamos esta información e intentamos explicar las regularidades

o patrones existentes en ella. Mediante el razonamiento inductivo, el científico construye una regla o teoría que explica las relaciones entre las observaciones. En búsqueda de una teoría que explique las relaciones entre las observaciones, el científico selecciona unas observaciones de entre las disponibles, guiado por un cierto conocimiento subjetivo, de aquello que resulta de interés. La ciencia evoluciona porque observaciones que en un principio no se consideraron, se muestran más tarde como importantes o a la inversa.

El termino aprendizaje máquina apareció inicialmente ligado a la Inteligencia Artificial, campo en el que se desarrollan gran número de técnicas como las de Kohonen [7] denominadas de auto organización y autoaprendizaje, el método de aprendizaje en arboles binarios y trabajos iniciales en el perceptrón, que condujeron posteriormente a las redes neuronales. Simultáneamente se han ido produciendo avances del aprendizaje dentro del campo de la estadística, dando lugar a un aprendizaje estadístico. Actualmente se nota una tendencia convergente en ambos enfoques relativos al aprendizaje en general.

El problema del aprendizaje es sólo una parte del procedimiento seguido en las ciencias, la ingeniería, las ciencias sociales, la medicina, y en general en todos los campos que utilizan la estadística, para extraer conclusiones.

En la figura 2 se representa un esquema de procedimiento general para estimar o aprender un modelo de relación entre datos.

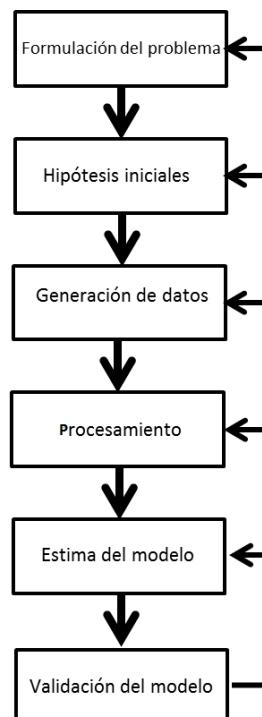


Figura 2: Procedimiento para aprender un modelo de relación de datos.



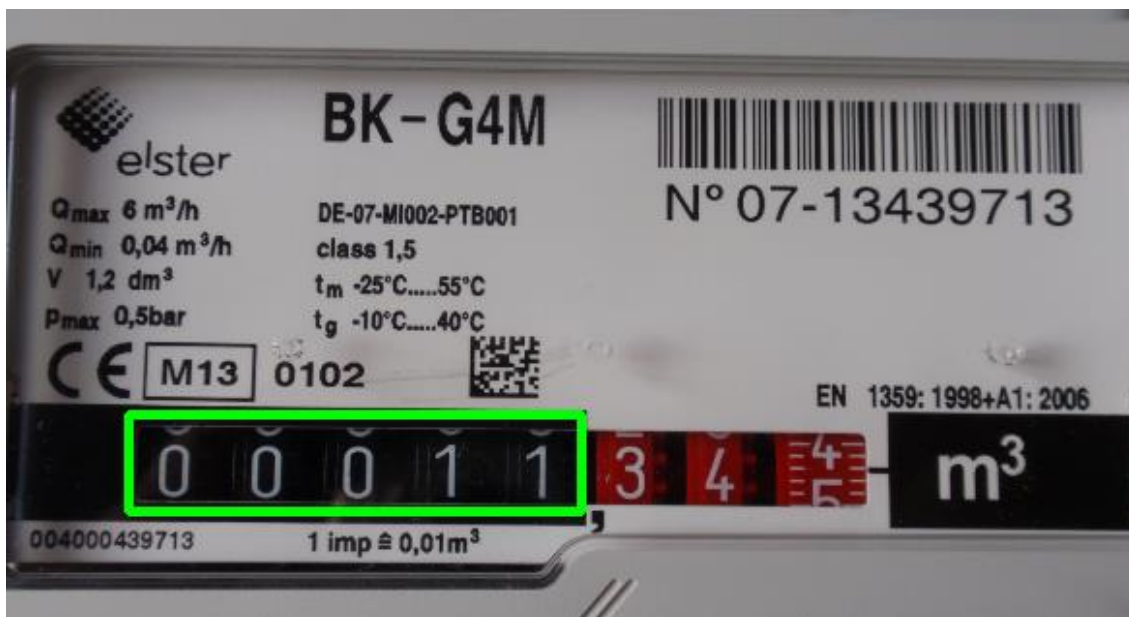
2.4 Reconocimiento Óptico de Caracteres (OCR)

Las siglas OCR son la abreviatura del reconocimiento óptico de caracteres (*Optical Character Recognition*), es un proceso dirigido a la digitalización de textos, los cuales son identificados automáticamente a partir de una imagen que contiene símbolos o caracteres que pertenecen a un alfabeto determinado, para después de ser reconocidos como secuencias de dicho alfabeto y ser posteriormente almacenados en forma de datos.

Las técnicas OCR constituyen una clara aplicación dentro del ámbito de aprendizaje automático, que sigue exactamente el modelo de esquema de éste, siendo a la vez una parte esencial del proceso de lectura de datos del contador propuesto en el presente trabajo. En efecto, siguiendo el esquema de dicha figura, el OCR trata de resolver el problema de automatización de la lectura de contadores e identificación de usuario bajo las hipótesis de partida consistente en la captura de imágenes del contador conteniendo la información mencionada. A partir de dichas imágenes se identifican las zonas de interés conteniendo tal información. Estas zonas son convenientemente procesadas mediante los métodos de tratamiento de imágenes explicadas posteriormente.

Dichas técnicas incluyen los métodos específicos de reconocimiento de caracteres, que permiten diferenciar unos caracteres de otros por comparación con los modelos disponibles y previamente almacenados durante la fase conocida como de entrenamiento del sistema dentro del proceso que forma parte del aprendizaje automático. La validación de los caracteres frente al modelo constituye la fase final de identificación de la secuencia de caracteres, que finalmente se traduce en la cadena de caracteres otorgándoles el valor numérico requerido para determinar el valor de la lectura y el identificador de usuario asociado.

En la figura 3 podemos apreciar la región de interés seleccionada bordeando los dígitos con la medición del contador. Esta región será la tratada por OCR.



.Figura 3: Reconocimiento de Caracteres imagen a tratar (OCR).

La figura 4 muestra el resultado de haber realizado OCR a la región de la imagen anterior en la aplicación.

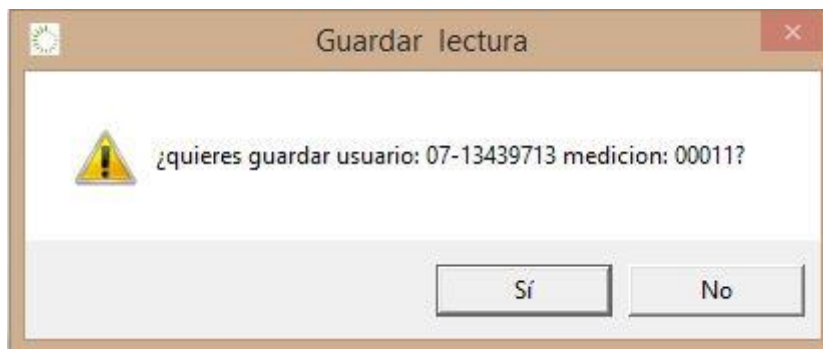


Figura 4: Resultado de aplicar OCR a una imagen.



CAPÍTULO 3: TÉCNICAS Y METODOLOGÍAS DE TRATAMIENTO DE IMÁGENES

En este capítulo se describen las técnicas y metodologías en el ámbito de la percepción computacional que se han utilizado para el tratamiento de imágenes y la obtención óptima de la información requerida en el proyecto.

Para conseguir una imagen más nítida y con mayor calidad a partir de la original, que en ocasiones puede presentarse con claras deficiencias de calidad, se han aplicado métodos de realzado a las imágenes originales. Una vez obtenida la imagen en la aplicación, utilizando las librerías de Python, se ha utilizado la función ROI para recortar la parte de interés de la imagen dejando solamente la secuencia numérica del registro correspondiente a la lectura del contador y la identificación del usuario.

Posteriormente, sobre estos recortes, contenedores de los dígitos de la región de interés, se aplicó una Binarización para conseguir una imagen con sólo dos valores numéricos, que en términos de imagen se corresponden con blancos y negros.

Finalmente, se aplicó un método de tratamiento conocido como apertura morfológica para limpiar el fondo en negro de la imagen de pequeñas marcas blancas que pudieran entorpecer el proceso de OCR.

Estos algoritmos son aplicados a las imágenes para garantizar una tasa de acierto mayor en el reconocimiento de caracteres numéricos, eliminando ruidos en las zonas más sensibles para nuestros intereses.

3.1 Análisis de Imágenes

Las imágenes digitales básicas en color están formadas por tres componentes espectrales correspondientes a cada uno de los canales R (rojo), G (verde) y B (azul) en el modelo de color RGB (Red, Green, Blue). El modelo RGB es un modelo basado en la síntesis aditiva, con el que es posible representar un color mediante la mezcla por adicción de los tres colores de luz primarios.

Cada canal R, G y B se representa mediante una matriz cuyos elementos, en una posición determinada (x,y), son los píxeles tomando valores numéricos en un rango determinado, normalmente entre [0, 255] cuando la representación de cada píxel es de 8 bits. Estas imágenes en color se pueden transformar convenientemente a diferentes modelos de color, entre los que se encuentra el

modelo HSV, también con tres componentes, donde la componente V es la de intensidad o imagen de gris, en este caso se trata de una única matriz de gris con valores en el rango $[0,255]$ con la misma representación anterior.

Sobre la imagen de intensidad resulta habitual realizar determinado tipo de operaciones o transformaciones que generan imágenes con dos únicos valores 0 ó 255, es decir valores negro o blanco respectivamente, que desde el punto de vista lógico se corresponden con valores 0 y 1. Estas imágenes se conocen como imágenes binarias que se obtienen aplicando técnicas conocidas como de Binarización, basadas en la aplicación de un umbral, cuyas técnicas se conocen como técnicas de umbralización o *thresholding* [7].

3.1.1 Imagen de Intensidad

A partir de las imágenes en color se obtiene la correspondiente imagen de intensidad, mediante la transformación de la imagen en el espacio de color RGB (Red, Green, Blue) al HSI (Hue, Saturation, Intensity) (Pajares y Cruz 2007), de suerte que es precisamente la de intensidad la utilizada en los procesos posteriores relativos a su tratamiento.

La figura 5 muestra un ejemplo de la imagen de intensidad de los dígitos correspondientes a lectura del contador.



Figura 5: Tratamiento de imagen de intensidad sobre imagen del contador.

3.1.2 Métodos del valor umbral

Los métodos del valor umbral son un conjunto de algoritmos que tienen como finalidad tratar una imagen para conseguir de ella sólo los elementos de interés, descartando el resto. De esta forma, en las imágenes con características más simples se puede determinar con mayor facilidad qué píxeles conforman los mencionados objetos de interés en la imagen y cuáles se pueden rechazar. Tras la aplicación de estos métodos se está en disposición de aplicar el OCR para la identificación propiamente dicha de los caracteres relativos a la lectura del contador y la identificación de usuario. A continuación

se describen varios métodos utilizados en el presente proyecto para llevar a cabo los procesos de Binarización por umbral.

La umbralización [7] es uno de los métodos de segmentación más importantes que existen. El objetivo es convertir una imagen de intensidad a una nueva con únicamente dos niveles, de manera que los objetos queden separados del fondo.

La técnica de umbralización consiste en un proceso de variación de la información de la imagen reduciendo los valores de la imagen a dos los negros y blancos antes mencionados. Lo que se hace es comparar cada píxel de la imagen con un determinado umbral para determinar si el píxel que estamos observando pertenece al negro (tendrá un valor por debajo del umbral) o al blanco (valor por encima del umbral).

Dentro de la propia umbralización existen distintos métodos, algunos de ellos incluso automáticos, como el de Otsu [20]. Estos métodos se usan dependiendo de la imagen a tratar, los distintos tratamientos son los que se presentan en los siguientes puntos.

En el tratamiento de imágenes utilizado en el presente proyecto se aplica el método de umbralización por los motivos previamente descritos, ayudando a diferenciar el fondo de la imagen y facilitando el reconocimiento de los caracteres numéricos.

La umbralización a cero es una técnica que consiste en que los píxeles por encima del valor del umbral mantienen el valor de la imagen origen y los valores por debajo del valor umbral se ponen a 0 (negro).

También existe la técnica de umbralización a cero invertida, en la que los valores que son superiores al valor umbral se ponen a 0 (negro) y los valores por debajo se mantienen a su valor de la imagen origen.

La figura 6 representa el número de medición del contador tras aplicar la umbralización a cero sobre la imagen original.



Figura 6: Tratamiento de imagen de umbralización.

3.1.3 Apertura Morfológica (Opening)

Apertura morfológica [29] es un tratamiento específico de imágenes que como su nombre indica se encuadra dentro de las operaciones morfológicas aplicadas a una imagen, teniendo como objetivo mejorar la calidad de la imagen binarizada, eliminando partes no deseadas procedentes de los procesos anteriores.

En este proyecto se ha utilizado la transformación de apertura morfológica que consiste en aplicar una dilatación seguida de una erosión. El resultado de este tratamiento sobre la imagen es la eliminación de pequeños detalles fuera de los objetos de interés para que no distorsionen de forma general el proceso de identificación de caracteres. Es decir, una vez binarizada la imagen disponiéndola en blanco y negro, la apertura morfológica lo que hace es eliminar pequeños puntos blancos sobre negro o a la inversa, minimizando así los errores durante la aplicación del OCR.

En la figura 7 se muestra el resultado obtenido tras la aplicación de un proceso de apertura binaria.



Figura 7: Tratamiento de imagen apertura morfológica sobre imagen del contador.

Todas estas técnicas son vitales para el desarrollo del presente proyecto, ya que permiten simplificar las imágenes originales y mejorar su calidad para facilitar el trabajo de reconocimiento de caracteres. En el punto cuatro de este documento, se detalla todo el proceso de tratamiento que recibe una imagen antes de ser enviada al módulo de OCR.



CAPÍTULO 4: ANALISIS Y DISEÑO

En el siguiente capítulo se aborda el análisis y diseño software, es decir la creación de la estructura lógica de la aplicación. Como se ha indicado previamente, se decidió utilizar Python y Mongo DB, de suerte que la implementación del programa va a estar dirigida por el patrón MVC (Modelo-Vista-Controlador). A continuación se describe la especificación de casos de uso principales y se desglosan dichos casos en los correspondientes subcasos.

4.1 Especificación Casos de Uso

Para cumplir con las funcionalidades decididas en equipo, según lo acordado en las diferentes reuniones de seguimiento del proyecto, se tienen que especificar los casos de uso necesarios con tal propósito, al objeto de obtener una máxima operatividad de cara al usuario final.

La aplicación de escritorio se divide en tres casos de uso principales, entendiendo como caso de uso principal una de las tres opciones que el usuario tiene disponibles el menú que se muestra cuando se inicia la interfaz al ejecutar la aplicación.

Para la visualización de los casos de uso, se ha representado cada uno de ellos con un diagrama de flujo o actividad, que permite mostrar de manera sencilla los procesos internos realizados. Posteriormente se desglosa individualmente cada caso en subcasos, de manera que se muestra el diagrama de flujo con divisiones, indicando en cada división, el subcaso de uso correspondiente de manera clara.

El primer caso de uso principal es el **procesamiento de la imagen**, que comprende la carga de ésta, la selección del área a recortar donde se encuentran los dígitos para la lectura del consumo o la identificación del cliente, el tratamiento del recorte, el reconocimiento de los caracteres y finalmente la confirmación por parte del usuario de los datos resultantes de la lectura para determinar si ésta es correcta, en cuyo caso se procederá al guardado de la información en la base de datos.

En la figura 8 se puede apreciar el diagrama de flujo que representa el caso principal del programa, tratamiento de la imagen.

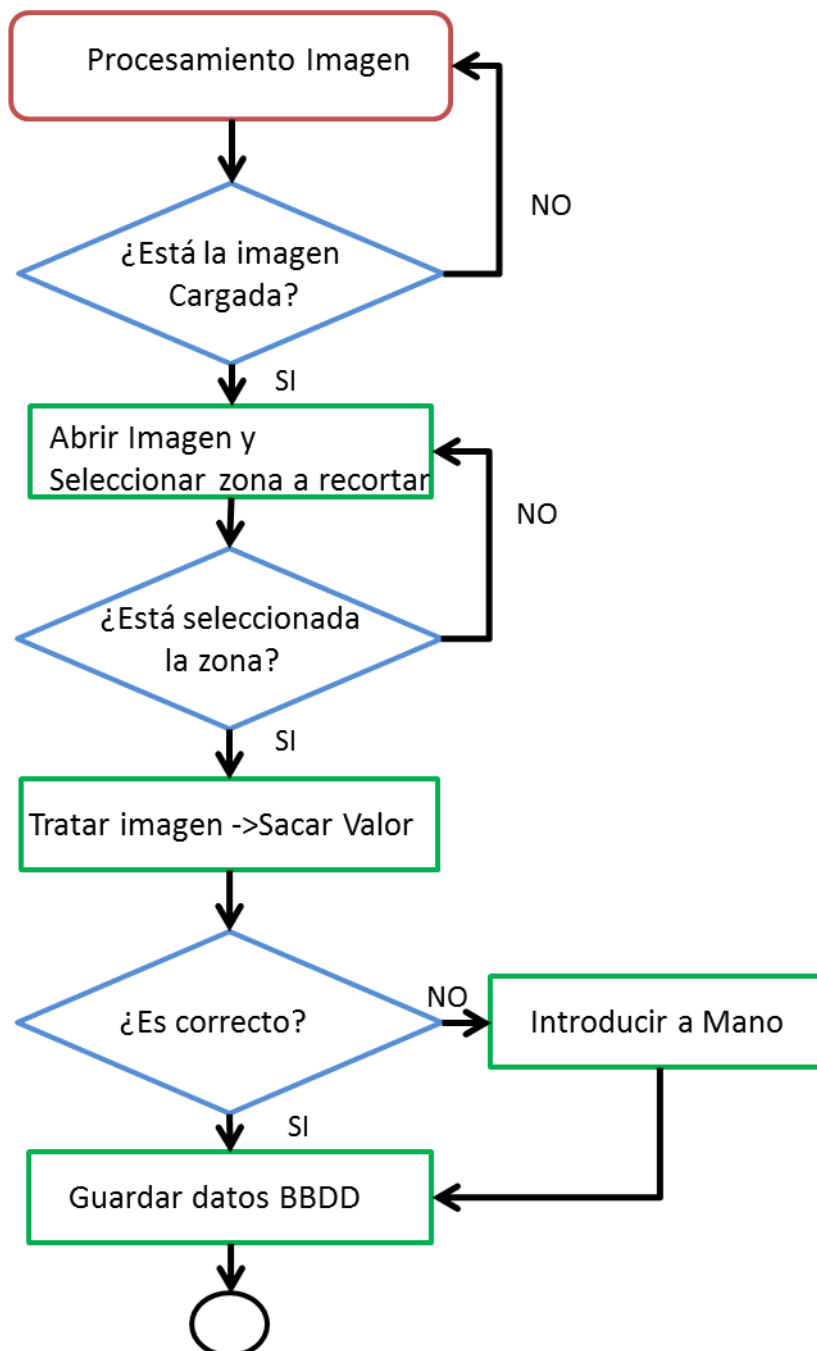


Figura 8: Diagrama de flujo para el caso principal procesar imagen.

El segundo caso de uso principal es el **histórico de cliente**, que comprende la visualización de todas las mediciones almacenadas para un cliente determinado. Primero habrá que comprobar que dicho cliente existe y una vez comprobado, se procede a mostrar la información de todas sus mediciones y sus fechas de proceso.

La figura 9 es un diagrama de flujo explicativo sobre el caso de uso de histórico del cliente.

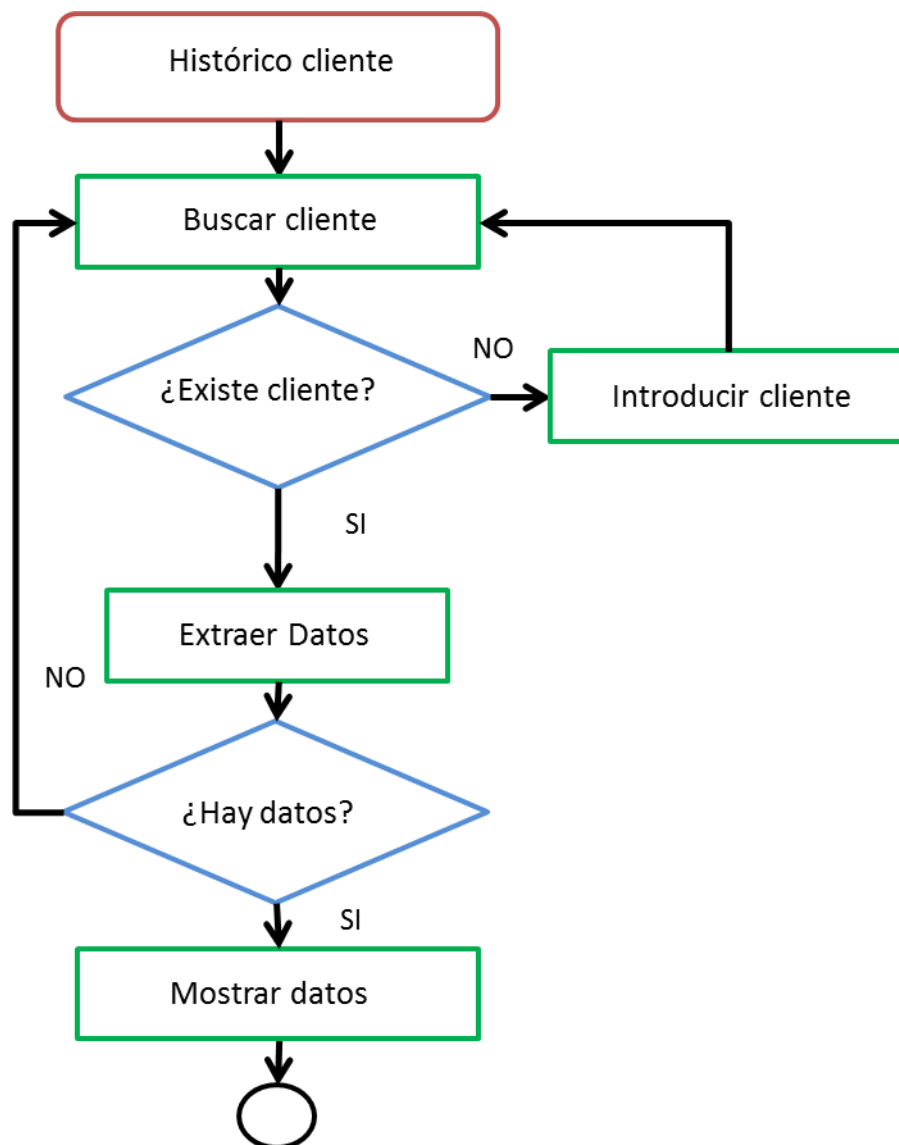


Figura 9: Diagrama de flujo para el caso principal Histórico del cliente.

El tercer y último caso de uso principal es el **listado de clientes**, que comprende la visualización de un listado con todos los clientes que tienen información almacenada en la base de datos.

La figura 10 representa el diagrama de flujo correspondiente al caso de uso de listado de cliente.

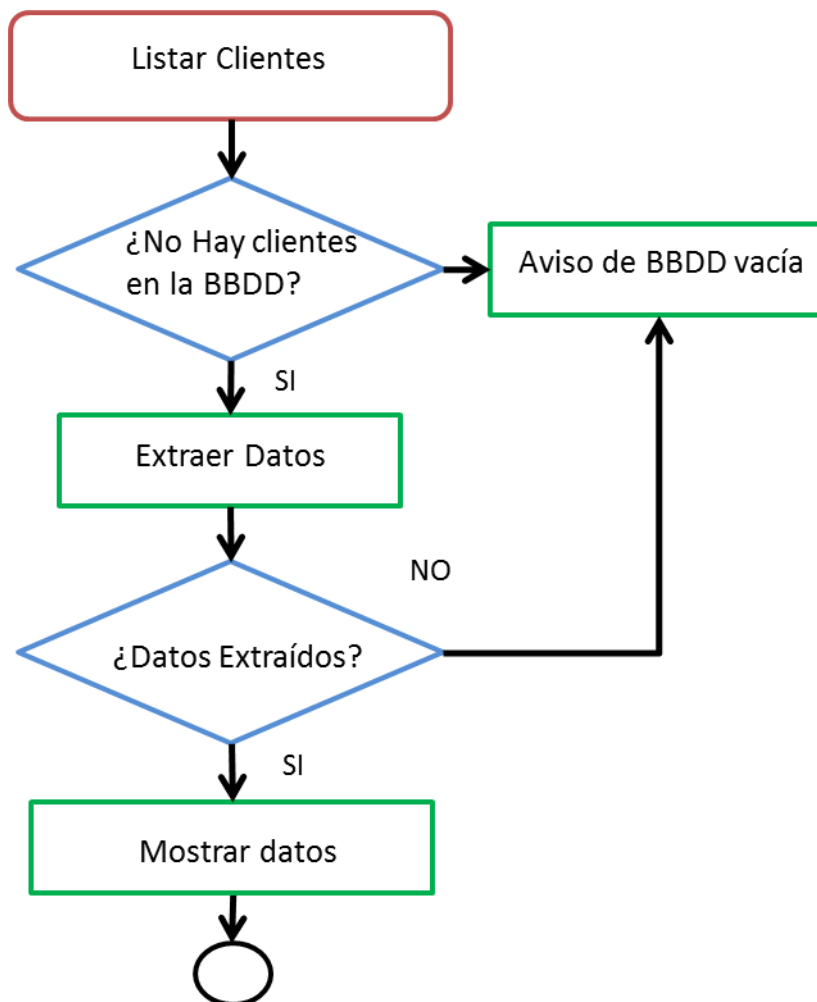


Figura 10: Diagrama de flujo para el caso principal listado de cliente.

Cada uno de estos tres casos de uso principales está compuesto de subcasos de uso más simples, que se detallan a continuación.



4.1.1 Caso de Uso principal: Procesar imagen

El caso de uso principal de tratamiento de imágenes contiene cinco subcasos de uso que se encuentran ubicados entre la vista y el modelo y que son seleccionados por el usuario de la aplicación de forma invisible para éste, y a su vez ejecutados por la aplicación. A continuación se explica cada subcaso de uso y su situación.

- **Cargar imagen**

Este subcaso de uso es ejecutado en la vista del programa por el usuario, que selecciona una imagen guardada en el ordenador y la dispone para su procesamiento, por lo que sería un subcaso de uso externo a la aplicación.

- **Recortar imagen**

En el subcaso de uso recortar imagen el usuario selecciona desde el exterior de la aplicación manualmente el área a recortar. El programa ejecuta el corte y devuelve al usuario la correspondiente imagen recortada.

- **Procesar imagen**

Se realiza internamente, siendo el encargado de aplicar a la imagen ya recortada todos los métodos de tratamiento de imagen descritos previamente, para obtener a continuación mediante OCR la correspondiente secuencia de caracteres con el identificador del cliente o la lectura del contador.

- **Confirmación de datos**

En el cuarto subcaso del caso de uso principal de tratamiento de imágenes se confirma que los datos extraídos del subcaso previo son correctos. Es de tipo externo ya que es el usuario del programa quien debe aceptar los datos extraídos de la imagen o en su defecto introducir el número correcto manualmente, para su correcta inserción en el subcaso de uso siguiente.

- **Guardar datos**

Una vez obtenida la confirmación por parte del usuario, el dato llega a la Base de Datos de suerte que ésta procede a almacenarlo. Una vez realizada esta operación se muestra un mensaje al usuario.

En la Figura 11 se muestra en un diagrama de flujo los subcasos de uso dependientes del caso principal procesar imagen explicado previamente.

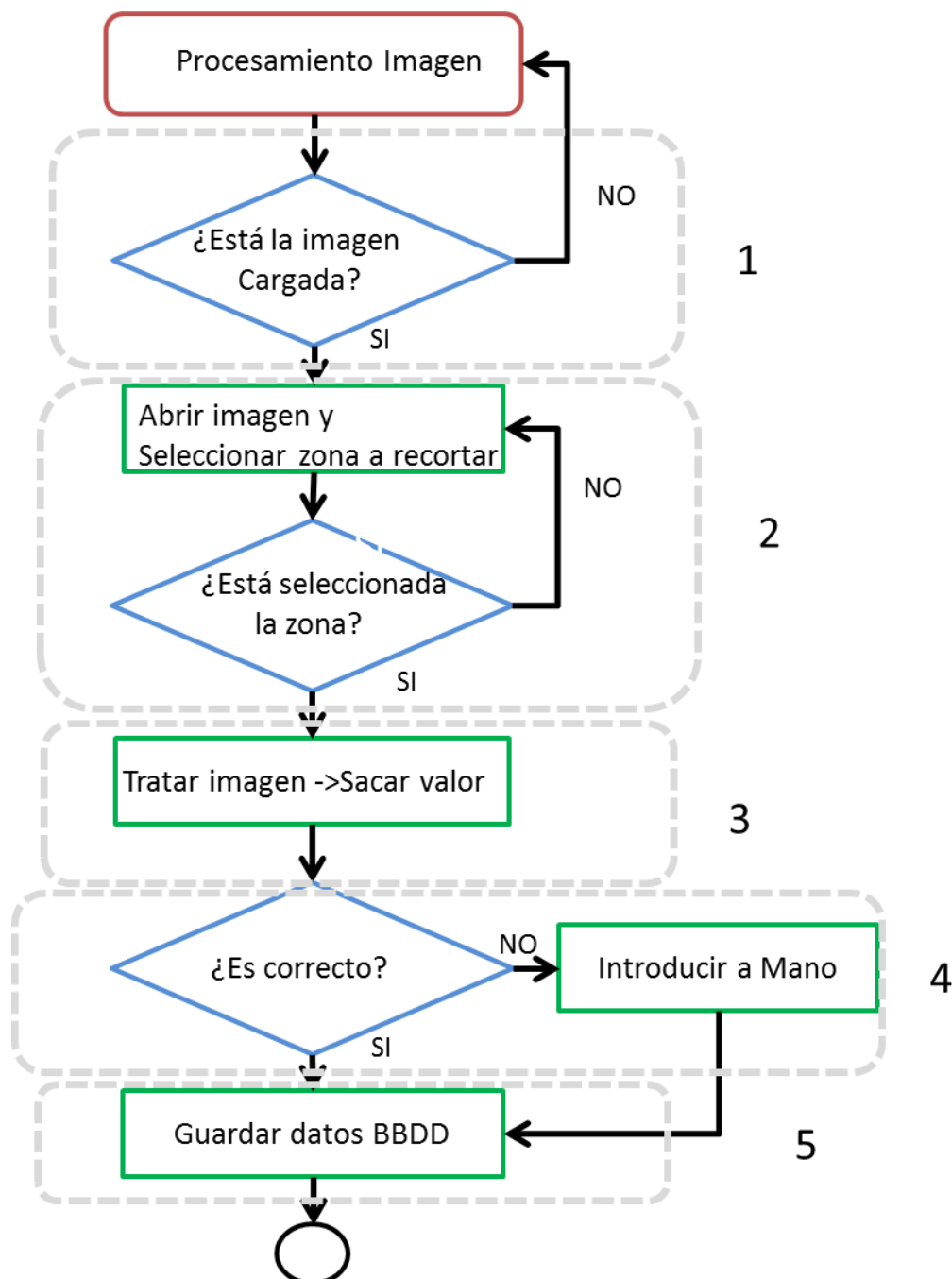


Figura 11: Diagrama de flujo del Caso de uso principal procesar imagen desglosado por subcasos de uso.



4.1.2 Caso de Uso principal: Histórico de cliente

El segundo caso de uso principal es el histórico de clientes. En él, se busca un código de cliente en la base de datos y se muestran todas las mediciones almacenadas acompañadas de la fecha en la que se procesaron.

Este caso de uso principal se desglosa en tres subcasos de uso, que se describen a continuación.

- **Buscar cliente en la Base de Datos**

Se trata de un subcaso externo ya que es el propio usuario de la aplicación quien debe aplicarlo. En primer lugar se introduce un código de cliente a través de la interfaz de usuario y a continuación se busca dicho cliente por su identificador en la BD. Si el cliente no existe la aplicación devuelve un mensaje aviso informando del error; en caso contrario se continúa con el subcaso siguiente.

- **Extraer datos de la Base de Datos**

Se trata de un subcaso interno, ya que se ejecuta de forma independiente con respecto al usuario. Una vez seleccionado el cliente y la información asociada, se invoca desde el modelo del programa a la base de datos para solicitar la información asociada a los datos suministrados. La base de datos devuelve dicha información al modelo para que éste la envíe para ser mostrada en el subcaso siguiente.

- **Mostrar datos extraídos**

Este subcaso de uso, también de tipo interno, se encarga de enviar a la interfaz gráfica de usuario la información extraída desde la base de datos para que el usuario de la aplicación pueda visualizarla. Esta información consiste en las mediciones de consumo del cliente seleccionado y la fecha de proceso en el sistema de esa medición en formato dd/mm/yyyy.

En la figura 12 se visualiza el caso de uso principal de histórico de cliente desglosado en los subcasos de uso explicados previamente.

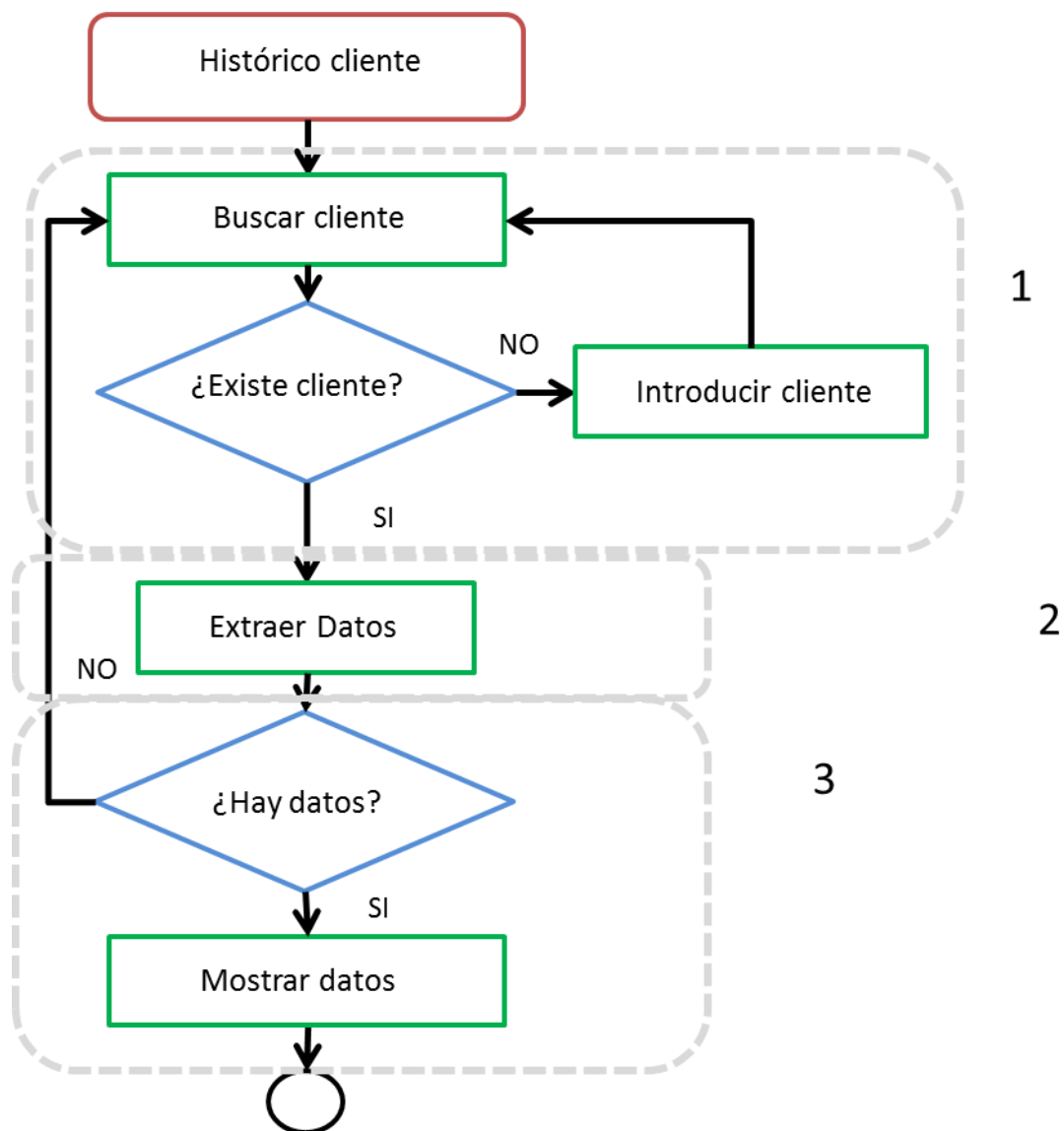


Figura 12: Diagrama de flujo histórico de cliente desglosado por subcasos.

4.1.3 Caso de Uso principal: Listado de clientes

El tercer y último caso de uso principal es el listado de clientes, cuya finalidad es seleccionar, listar y mostrar por pantalla todos los clientes almacenados en la base de datos, con su correspondiente identificador de usuario. Este subcaso se desglosa a su vez en tres subcasos de uso que se explican a continuación.

- **Buscar clientes en la Base de Datos**

El primer subcaso de uso que aparece dentro del flujo principal consiste en buscar los clientes, mediante una consulta, en la base de datos para mostrarlos. En caso de que no exista ningún cliente almacenado, se envía un mensaje de aviso a la interfaz gráfica de usuario del programa indicando al usuario que la base de datos está vacía. Se trata de un caso interno a la aplicación.

- **Extraer datos**

Se trata de un subcaso de uso de tipo interno que, una vez encontrada la información, extrae la información desde la base de datos y la ordena para enviarla a la interfaz gráfica de usuario. En caso de error durante el proceso de extracción de la información, se envía un mensaje de error al usuario informando del imprevisto.

- **Mostrar datos**

Este subcaso de uso, también de tipo interno, se encarga de mostrar la información extraída de la base de datos y enviada a la Vista desde el subcaso de uso anterior para su visualización por parte del usuario. El resultado es un listado con todos los clientes procesados por la aplicación ordenados por su código de identificación.

La figura 13 muestra una representación en diagrama de flujo del caso de uso principal de listado de clientes explicado previamente desglosado en los subcasos que contiene.

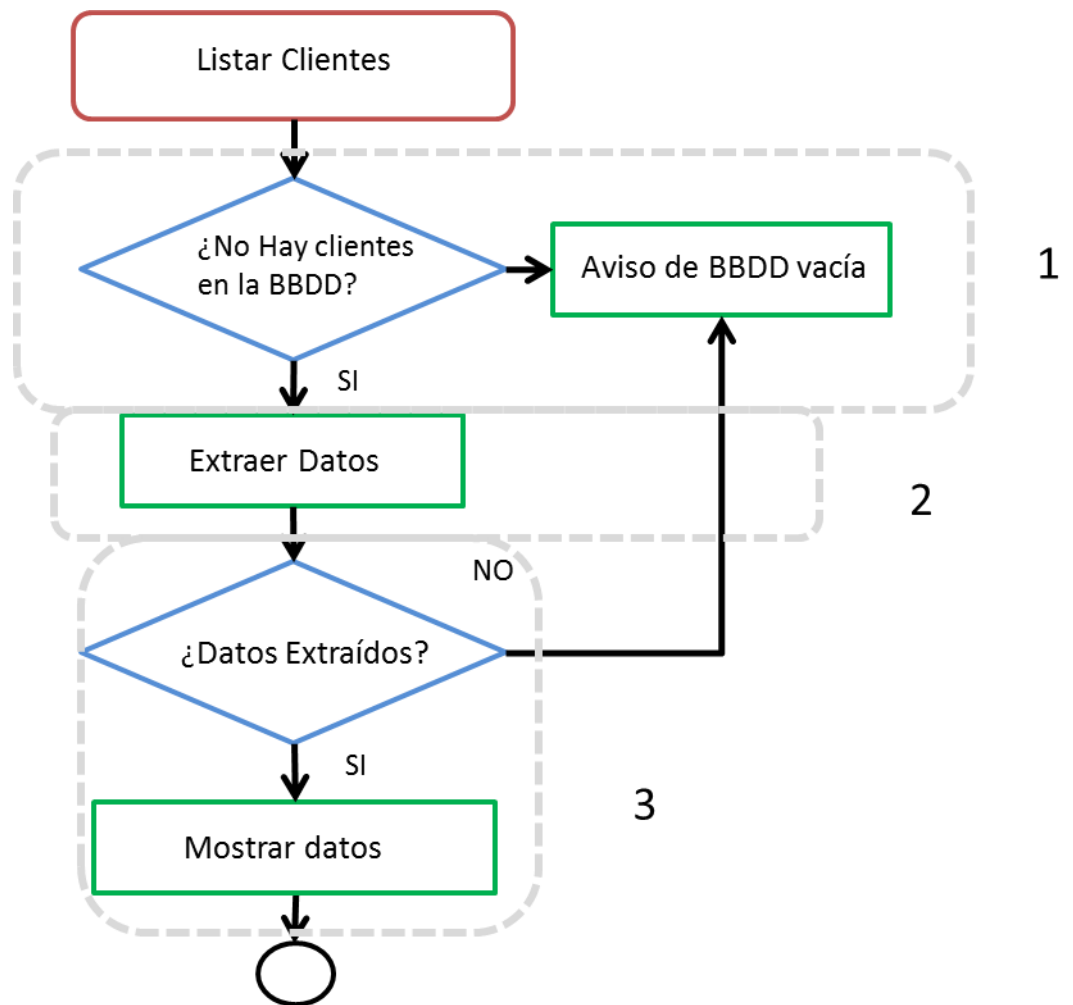


Figura 13: desglose del caso de uso principal listar Cliente por subcasos de uso.

4.2 Diseño: Modelo–Vista–Controlador

Para el diseño interno del software de la aplicación desarrollada se ha decidido utilizar para su implementación un patrón que cumpla con la organización prevista para la implementación, el patrón que se ha utilizado es el patrón Modelo-Vista-Controlador, también conocido como MVC [19] (Model-View-Controller). La estructura de este modelo viene dada por una parte lógica (Modelo), la parte visual o interfaz (Vista) y del núcleo de la aplicación (Controlador).

La figura 14 muestra en un esquema simple la relación que tiene el patrón entre las tres capas que lo componen y la respuesta que tiene con el usuario.

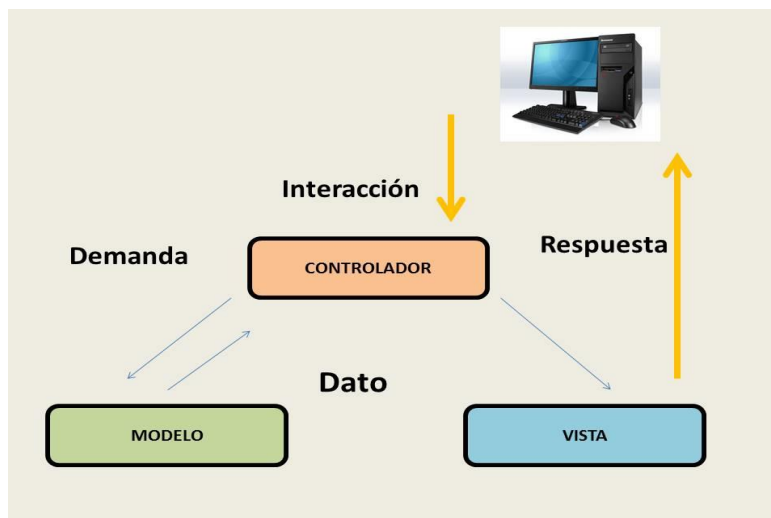


Figura 14: Estructuración Modelo-Vista-Controlador.

A continuación se exponen cada una de las tres capas de que dispone el patrón MVC para desarrollarlas con más detalle seguidamente.

- **MODELO:**

El modelo [19][32] es la parte del programa que implementa la lógica, sus responsabilidades dentro de la aplicación es la de interaccionar con la Base de Datos para obtener o insertar datos en ella y operar con ellos convirtiéndolos en conceptos con significado para el programa.

- **VISTA:**

La vista es la encargada de ejecutar una presentación de los datos que el modelo ha conseguido obtener, si bien, la vista está separada de los objetos del modelo. Es la encargada a su vez de producir mediante la interfaz de usuario cualquier petición que este haga.

- **CONTROLADOR:**

El controlador es el encargado de gestionar las peticiones enviadas desde la vista por el usuario de la aplicación, es un intermediario entre el modelo y la vista. El controlador es el administrador del programa que va equilibrando los recursos que necesitan ambas capas para completar su tarea.



El patrón MVC aporta grandes beneficios al programa que lo usa en su implementación, uno de ellos es su estructura modular, esto hace que sea fácil de mantener y modificar, aumentando la rapidez del desarrollo del programa; otro de los beneficios es el diseño de una aplicación más sencilla de diseñar y por tanto de entender, ya que separa muy claramente las peticiones de las operaciones y de la iteración con la base de datos y por ultimo una de las características que le hacen más interesante es la facilidad de modificación que posee, pudiendo agregar o eliminar funciones sin modificar el resto de las operaciones, lo que le otorga una gran flexibilidad y reusabilidad.

4.2.1 Funcionamiento MVC:

En este apartado se explica con un mayor nivel de detalle la interacción entre capas para el correcto funcionamiento del patrón MVC [12].

Es importante tener clara la idea de que el controlador es la capa donde se interpretan las peticiones del usuario de la aplicación y el encargado de enviar el mensaje con la acción al modelo para que la ejecute. Una vez ejecutada será la vista la que devuelva la petición con el dato extraído de la ejecución.

Ambas capas del patrón deben comunicarse claramente y entenderse entre sí de una manera perfecta, de forma que tengan entre ellas una coherencia en las iteraciones para el correcto funcionamiento del software.

Dentro del propio MVC pueden existir diferentes asociaciones, no necesariamente debe ser uno a uno entre modelo y vista o controlador, así un modelo puede estar asociado a varias vistas y controladores.

Al contrario, una vista va a tener una relación uno a uno con el controlador, cada vista va a estar relacionada con un único controlador, aunque compartan modelo con otras vistas asociadas a sus respectivos controladores.

Resumiendo la vista es la encargada de establecer la comunicación entre el usuario y el ordenador, avisándole a través del controlador al modelo que es el que ejecuta la acción requerida por el usuario y devolviendo dicho resultado a la vista para que sea reproducida.

4.3 MVC en el proyecto

Para tener una visión más explícita y concreta de cómo está constituida la implementación del software del proyecto, en este apartado se hace referencia a cada una de las partes que constituyen el patrón MVC (Modelo-Vista-Controlador) en el propio proyecto.



4.3.1 Asociación Modelo–Vista–Controlador

En el desarrollo que se plantea, la asociación entre modelo, vista y controlador viene estructurada como una relación uno a uno para todas las posibles asociaciones existentes entre sí. A continuación se describe detalladamente cada una de ellas.

- Asociación Vista-Controlador

En la aplicación se dispone de una única vista (Interfaz gráfica de usuario) asociada a un único controlador, el cual como ya se ha comentado, será el encargado de recibir las peticiones de la vista y enviar las operaciones a ejecutar al modelo.

- Asociación Vista-Modelo

La asociación entre la vista y el modelo también es uno a uno, sólo se dispone de una vista y de un modelo. De este modo, la vista envía la información de las operaciones que el usuario quiera realizar al controlador y el controlador sólo dispone de un modelo donde se encuentran dichas operaciones para ser ejecutadas.

- Asociación Modelo-Controlador

Como se describe en los puntos anteriores, tenemos un único controlador y un único modelo de manera que, el controlador sólo podrá enviar y requerir datos al modelo existente y éste solo podrá enviarlos a la vista a través del controlador.

4.3.2 Estructura e implementación en MVC

Modelo

El propósito principal del modelo es aplicar las fases de aprendizaje automático que tienen que ver con el tratamiento y obtención de información de la imagen, *figura 9*, es decir, el modelo recibe la imagen que el usuario ha seleccionado y se encarga de procesarla, y devolver los resultados obtenidos. El flujo de información que tiene lugar en el proceso discurre de la siguiente manera:



- El modelo recibe una ruta donde la imagen está almacenada en formato digital, devolviéndola preparada para su procesamiento.
- Una vez cargada la imagen, mientras el usuario selecciona una región desde la vista, el modelo se encarga de reconocer esa región y enviarla a la vista para mostrarla con un recuadro verde.
- A continuación, recibe el recuadro de la región de interés seleccionada y se encarga de crear una nueva imagen a partir de esa región.
- Recibe esta nueva imagen con la región seleccionada y es el encargado de realizar las operaciones de tratamiento de imágenes necesarias antes de aplicar el OCR [3], que devuelve una cadena de caracteres (string) con la medición del contador o el identificador de usuario.
- Dicha cadena string es tratada convenientemente en el modelo para comprobar que no haya espacios en blanco entre dígitos, ni recibamos caracteres “O” en lugar de ceros ya que solo se procesan dígitos numéricos.

Implementación del modelo

En relación a la parte de implementación, cabe distinguir entre las capas de integración y de negocio como se describe a continuación.

Capa Integración

Clase MONGODB

Contiene la base de datos mongoDB [13]. Utiliza la librería pymongo para conectarse a ella con los atributos base de datos y colección. Contiene métodos para insertar medición, obtener todas las mediciones a partir de un usuario y obtener todos los usuarios de la base de datos.

Capa Negocio

Clase utils

Utiliza las librerías OpenCv [5] y numpy [2]. Contiene los métodos relacionados con el tratamiento de una imagen y métodos auxiliares como transformar a ceros, abrir imagen...).

Clase cortarImagen

Efectúa las operaciones de recorte de la imagen, contiene los métodos para pintar el rectángulo de la región de interés, y otro método que devuelve un recorte, entendiendo por recorte una imagen nueva a partir del rectángulo que rodea la región de interés.



Clase TesseractOCR

Clase importada de un repositorio Github [36], contiene todos los métodos necesarios para la de la lectura de caracteres de una imagen optimizados para OpenCV.

Vista

El propósito de la vista es mostrar de manera amigable al usuario el entorno y el resultado de las operaciones realizadas por el modelo teniendo como intermediario entre ellos al controlador. Los trabajos que ejecuta la vista de la implementación son:

- Al iniciar el programa la vista muestra una ventana con el menú principal, que es la encargada de permitir que el usuario pueda visualizar la interfaz durante toda la ejecución.
- La vista permite que se realice el recorte de la imagen que posteriormente va a ser procesado por el modelo.

Implementación de la vista

La vista contiene una clase por cada una de las ventanas que se van a mostrar a través de la interfaz de usuario. Son las siguientes:

Clase Ventana

Inicializa la ventana principal con la imagen corporativa, la barra de menús y el contenedor donde se insertaran las diferentes ventanas. Al iniciar la aplicación se inserta en el contenedor la página de inicio.

Clase StartPage

Contiene la portada de la pantalla inicial.

Clase Contador

Contiene los elementos GUI para mostrar la imagen del contador, las lecturas y los botones para interactuar con la aplicación.

Clase usuario

Contiene los elementos GUI para mostrar la tabla con los usuarios y la tabla con las mediciones y fechas de procesamiento para un usuario.



Controlador

La función del controlador es gestionar el flujo de información entre modelo y vista haciendo independientes los dos módulos. Importa todas las clases de integración, negocio y presentación.

Implementación del controlador

Clase controlador

Contiene diferentes procedimientos que llaman a las funciones de las clases del modelo y la vista. Incluye también un método *process* que maneja los eventos y llama a los procedimientos anteriores en función de las peticiones que se van realizando desde la vista.

4.4 Arquitectura Software

Una vez descrita la arquitectura interna del software y su distribución, a continuación se explica el esquema general de lo que realmente resulta ser el comportamiento de la funcionalidad software basando dicha funcionalidad en los casos de uso explicados en el punto 4.

Cuando el usuario lanza el programa aparece en la ventana principal (interfaz gráfica) un mensaje indicando el comienzo. La interfaz es una ventana simple y sencilla que cuenta con botones en la parte superior donde el usuario tiene la opción de elegir el caso de uso a ejecutar, además de poder salir de la aplicación si lo desea.

Si el usuario elige la opción del reloj el usuario entraría en el caso de uso relativo al tratamiento de imágenes, en el cual lo primero que deberá hacer es seleccionar la opción lateral izquierda de *Abrir Imagen* para seleccionar una imagen desde su escritorio y después proceder a pulsar la opción *Encuadrar Imagen* para realizar los recortes necesarios con el fin de reducir la región de lectura o la de identificación del cliente. Estas regiones constituyen el objetivo del tratamiento de imágenes propiamente dicho. Aceptados los recortes, por parte del usuario, se procede a utilizar la opción de *Leer Usuario* si el recorte efectuado con anterioridad es la región del número de cliente o *Leer medición* si lo que se va a interpretar es la medición de la lectura del contador de gas. Una vez el programa muestra su interpretación de los datos, el usuario procede a seleccionar *Guardar Datos* si está conforme con lo interpretado o introducirlos y a continuación “*guardar datos*” en caso contrario. Una vez el programa emita

un mensaje de que los datos han sido guardados satisfactoriamente, habrá finalizado la opción correspondiente al caso de uso de tratamiento de imágenes.

Si por el contrario el usuario elige la opción representada con el símbolo de una hoja de papel relativo al tratamiento de imágenes, se activaría el caso de uso del histórico del cliente, de forma que si el cliente ya ha sido identificado nos mostrara el histórico del cliente que se tenga en ese momento identificado, si no hubiera cliente, habrá que introducirlo a mano y a continuación aceptar para que el programa devuelva el listado con el histórico del cliente deseado.

Por último, si se decide seleccionar la opción de los iconos con forma de humano el usuario entraría en el caso de uso de Listar clientes en el cual se le mostrará nada más pulsar el botón de la página principal un listado con todos los clientes que existen en la base de datos en la actualidad, si al pulsar el botón de la página principal no existieran clientes en la base de datos se le enviara una notificación al usuario diciendo que no hay clientes en la base de datos.

En la figura 15 se muestra un esquema ilustrativo del MVC aplicado al proyecto.

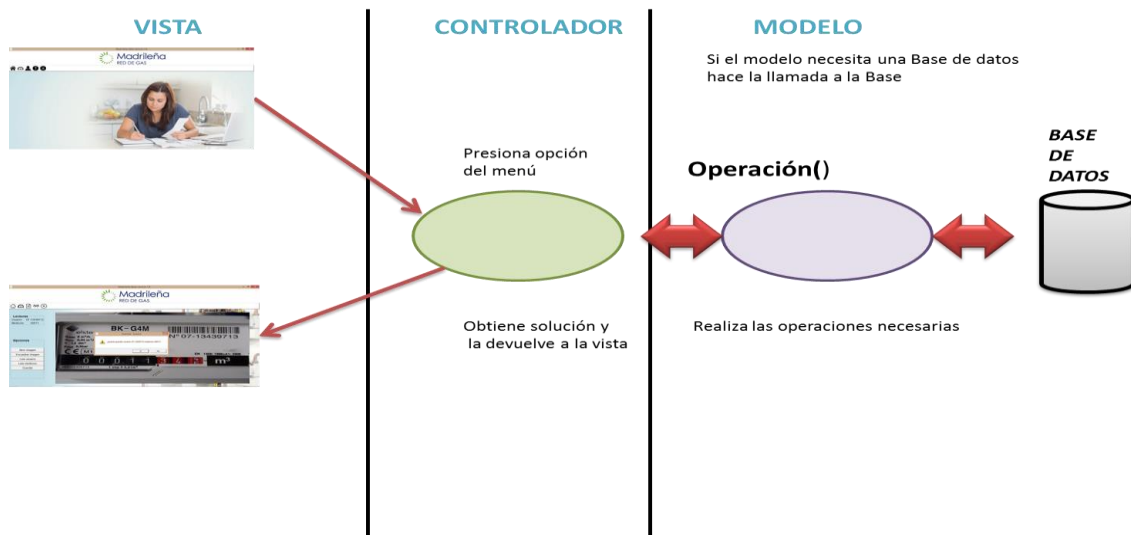


Figura 15: Enfoque de la implementación MVC.





CAPÍTULO 5: IMPLEMENTACION

5.1 Tecnologías utilizadas

Para la realización de la implementación del programa se han utilizado diversas tecnologías tales como librerías, compiladores, clases, programas, lenguajes de programación, todas ellas ya desarrolladas e implementadas y de libre acceso con disponibilidad ilimitada, a Las tecnologías utilizadas han sido seleccionadas por su comportamiento en las tareas de visión computacional, a su vez se han ido descartando del catálogo de tecnologías las que proporcionan rendimientos no acordes con los requisitos como puede ser Matlab o el lenguaje de programación C++.

El lenguaje de programación utilizado para el desarrollo de la aplicación es Python [24], tal y como se indicó previamente, dado que contiene en sus diferentes librerías los métodos requeridos en el presente proyecto ya implementados, además de disponer de múltiple documentación asociada, pudiendo tratar problemas más complejos que otros lenguajes no permitirían.

La decisión de desarrollar una aplicación en entorno de escritorio vino motivada por la facilidad de instalación en cualquier dispositivo y el cómodo mantenimiento que ello conllevaba, ya que no necesitaba conexión vía internet y sólo sería de uso exclusivo en la empresa, aunque se dejó abierta la puerta para una posible migración a plataforma web en un futuro y en siguientes versiones del proyecto.

Las tecnologías utilizadas se desglosan siguiendo la misma estructura de la implementación de modelo-vista-controlador.

Para la capa de presentación, al tratarse de una aplicación de escritorio, el propio lenguaje de programación Python con la librería específica para la implementación de interfaces que es Tkinter [21] [22] fue suficiente para dejar implementada una versión funcional y operativa. Esta versión de la interfaz se adapta de manera automática a cualquier tamaño de pantalla.

Para la capa del controlador, al ser éste un mero mensajero entre las capas de modelo y vista sólo se ha necesitado la tecnología del lenguaje de programación usado para la elaboración de la aplicación, Python.

Para la capa de modelo, es donde más tecnológicas fueron necesarias, ya que es la capa donde interactúa la base de datos y el tratamiento de imágenes.

Desde el punto de vista del tratamiento de imágenes la librería elegida fue OpenCV por ser una de las más potentes en relación a las necesidades tecnológicas del proyecto, ya que posee todas las técnicas mencionadas con anterioridad. Además de existir una documentación exhaustiva sobre ella que permitía ver ejemplos similares para poder captar las ideas necesarias para el correcto desarrollo del software. Dentro de OpenCv [5] las librerías que se han usado para la implementación son *numpy* [2] para el soporte de matrices y vectores y *matplotlib* para la visualización de los resultados.

Sobre el tratamiento de la imagen, otro punto importante a considerar era la necesidad de pasar de una imagen a texto después de aplicar los correspondientes tratamientos a la imagen, para tal propósito existen los OCR descritos con anterioridad, en este caso se usó Tesseract [7], adaptándose al lenguaje de programación de Python [8]. Se utiliza esta tecnología dado que es una de las más potentes existentes en el mercado en la actualidad.

Para la Base de Datos (BD), como ya se ha comentado previamente, se eligió DB MONGO por dos razones también comentadas. Por una parte, es la base de datos que menos conflicto causaba con el lenguaje de programación y además, para desarrollar el programa deseado de escritorio era la que mejor se adaptaba, además de que su manejo de datos es ciertamente novedoso y flexible. Para compatibilizar la BD con el lenguaje de programación se usó la librería *pymongo* de Python.

Finalmente, desde el punto de vista de la gestión del proyecto y para facilitar el trabajo en equipo durante el desarrollo y documentación del proyecto se han utilizado las siguientes tecnologías:

- Para el trabajo conjunto del proyecto se decidió usar la herramienta de google drive, que facilita la subida de documentación y código.
- Para la elaboración de la documentación se eligió MS office, que facilita el progreso con su correspondiente control de cambios

5.2 Proceso de reconocimiento de lecturas

En este apartado se describen brevemente los resultados obtenidos mediante el tratamiento de imágenes, ya que al ser un proceso invisible para el usuario, a la hora de cargar la imagen, no se pueden visualizar todos los pasos internos aplicados sobre la imagen hasta que ésta vuelve a ser visualizada por el usuario una vez procesada. En este caso se detallan los casos disponibles relacionados con el proceso interno de tratamiento de la imagen: recorte,

reconocimiento de caracteres de las zonas de interés de la lectura del contador y del área de identificación de usuario. A continuación se detallan los dos casos de reconocimiento de lectura.

5.2.1 Recorte del área seleccionada

El recorte del área seleccionada es el punto donde el usuario, una vez ha cargado la imagen seleccionada en la aplicación, elige dentro de la propia imagen completa el área de interés que mejor se ajuste al recuadro conteniendo la parte del contador donde se encuentran los dígitos objeto de su tratamiento. Se realiza un primer encuadre tal y como muestra la figura 16, para posteriormente proceder a enmarcar los dígitos correspondientes a la lectura del contador. Se excluyen de este encuadre los decimales por carecer de interés operativo desde el punto de vista de la lectura por parte de la empresa.

Para efectuar el recorte el usuario debe arrastrar con el ratón el recuadro verde que aparece sobre el área deseada.

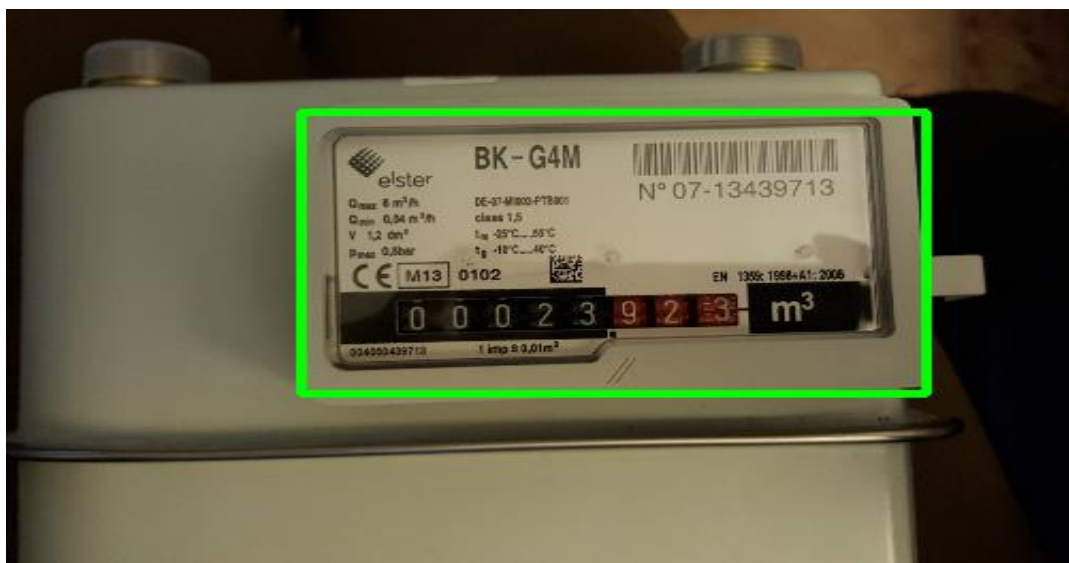


Figura 16: Primer encuadre del contador.

La Figura 17 muestra cómo se selecciona exclusivamente el área de interés para la lectura de la medición del contador.



Figura 17: Selección del área de medición del contador.

El recorte se efectúa de la misma manera tanto para el área de la lectura como para el área conteniendo el identificador del cliente. En la figura 18 se puede comprobar cómo se selecciona el área donde se sitúa la identificación del cliente para su reconocimiento.



Figura 18: Selección del área de identificación del cliente.

Conviene señalar que cuanto mejor encuadre se consiga, mejor serán los resultados, evitando así la inclusión de partes de la imagen irrelevantes que pudieran llevar a confusión a la hora de las identificaciones, este punto tiene vital importancia para obtener los resultados deseados.

5.2.2 Reconocimiento de los dígitos

El reconocimiento de lectura del contador comienza una vez se ha realizado el recorte del área seleccionada, como se puede ver en la Figura 19 ya sólo se dispone del área que el usuario de la aplicación ha escogido como área para que se reconozcan los caracteres, la zona más importante del recorte es donde se sitúan los dígitos de la lectura, que deben estar encuadrados perfectamente en el recorte. La figura 19 muestra el área seleccionada para la medición de la lectura del contador.



Figura 19: Área final recortada para el tratamiento de imágenes.

El reconocimiento de los caracteres para la lectura del gas comienza una vez se ha efectuado el recorte señalado anteriormente, el primer tratamiento aplicado a la imagen es el paso de la imagen RGB a la de intensidad mediante la correspondiente transformación de RGB a HSI. La figura 20 muestra el resultado de dicha transformación.



Figura 20: imagen con el tratamiento de intensidad.

Sobre la imagen de intensidad se aplica la Binarización para obtener sólo los dos valores binarios de interés, como se observa en la figura 21.



Figura 21: Resultado de aplicar el tratamiento de Binarización.

Por último, antes de aplicar el OCR para que traduzca los dígitos a una secuencia de caracteres, se aplica una apertura morfológica, para eliminar posibles imperfecciones o píxeles aislados que puedan causar confusión al OCR. En la figura 22 se puede apreciar cómo se modifican levemente los dígitos de interés.



Figura 22: Resultado de aplicar el tratamiento de apertura morfológica.

Una vez se han aplicado los tratamientos previos, el OCR es el encargado del siguiente paso de procesamiento. Es necesario resaltar en este punto, que el OCR es un método basado en técnicas de aprendizaje automático, concretamente métodos de agrupamiento borroso, Bayes, SVM y otros, de suerte que para lograr la máxima efectividad ha sido necesario realizar un proceso de entrenamiento previo, consistente en mostrarle los posibles caracteres con los que se va a encontrar posteriormente durante la fase de decisión. Concretamente se muestran los caracteres del 0 al 9 extraídos previamente de imágenes procedentes de contadores.

CAPÍTULO 6: CONCLUSIONES Y FUTURAS MEJORAS

El objetivo principal del presente proyecto era dar solución al problema planteado por la empresa Madrileña Red de Gas para automatizar en la medida de lo posible la lectura de consumo en los contadores de gas, así como la identificación de usuario a partir de imágenes diversas y variopintas capturadas por los propios clientes con sus propios dispositivos de captura de imágenes, tras su envío a la sede de la empresa.

La solución aportada ha conseguido su propósito de forma satisfactoria reduciendo la tasa de fallos en el procesamiento de imágenes en entornos reales gracias al sistema de encuadre y recorte manual y al tratamiento aplicado sobre las imágenes. De esta forma se consigue incrementar el proceso de lectura con su correspondiente identificador de usuario.

La aplicación también incluye un apartado de gestión de clientes donde se pueden consultar todas las mediciones de un cliente a lo largo del tiempo, lo que permite a Madrileña Red de Gas tener acceso en cualquier momento para poder comprobar si la línea de consumo mes a mes está siendo coherente con lo estimado.

La reducción de la tasa de fallos, acorde con las expectativas de la empresa, permitirá a Madrileña Red de Gas reducir costes, ya que gracias a la aplicación desarrollada, el número de técnicos que deben desplazarse a los domicilios de los clientes disminuye, y estos últimos no dependen de la visita de un técnico para poder enviar su consumo mensual.

Desde el punto de vista de una aplicación propia de desarrollo de software, los objetivos más importantes conseguidos en el desarrollo del proyecto son:

- Entregar en plazo una aplicación funcional acorde con los requisitos establecidos en la primera reunión.
- Reducir la tasa de fallos en el procesamiento de las imágenes.
- Poder procesar imágenes tomadas a mayor distancia gracias al sistema de encuadre manual.
- Implementar la aplicación bajo el patrón MVC para ser capaces de adaptar nuevas interfaces y otros sistemas de persistencia de la información de manera sencilla.



Las mejoras futuras que quedarían pendientes de desarrollo para próximas versiones de la aplicación serían:

- Mantener la tasa de fallos actual automatizando el sistema de procesamiento de imágenes.
- Continuar reduciendo la tasa de fallos utilizando nuevos métodos de reconocimiento óptico de caracteres.
- Incorporar un sistema de estimación de consumo utilizando el histórico de mediciones por si no se recibe la medición de un cliente algún mes
- Actualizar los lenguajes y librerías utilizadas para poder disponer de mejores herramientas.
- Adaptar la persistencia de datos para incorporar plataformas Cloud.

CONCLUSIONS AND FUTURE WORKS

The main objective of this project was to solve the automatic reading problem in gas meters images to determine the consumption at the same time to read the consumer's identification code. In this way, the possibility to solve the problem is provided and raised by the company Madrileña Red de Gas. The software-based application has been designed to work in images captured in real environments, even if under adverse illumination conditions. The company has put the emphasis on effectiveness at the expense of decrease the degree of automatization. In this regard, the process is based on a manual selection of the region of interest to be processed. In addition, a set of specific computer vision techniques together with machine learning approaches have been applied to achieve such goal. This increases the rate of success.

The reduction in the error rate according to the expectative by the company and the director of the project, allows to Madrileña Red de Gas costs reduction providing a better service to the costumers. Thus, thanks to the proposed approach, involving appropriate image processing and pattern recognition techniques, important benefits are obtained for costs reduction.

Finally, the application includes a management section supported by a Data Base, where all recorded records for clients can be consulted for consumptions. Madrileña Red de Gas has free access for estimations and control.

The main objectives are summarized as follows:

- The application fulfils the initial requirements.
- The error rate during character recognition has been minimized.
- Real images, acquired under different illumination conditions and resolutions, are processed with acceptable performances.
- The application is designed under the MVC paradigm with friendly interfaces and reusability performances.

Pending actions and improvements in the future can be summarized as follows:

- Increase the performance, minimizing the error rate by applying additional image processing techniques based on all available real cases.
- Apply new character recognition approaches to verify their performances against the proposed approach in this work.



- Include a new estimation procedure for the consumption; the historic records can be used for this action thanks to the use of the Data Base.
- Update language and tools for better performances.
- Progress toward the Internet of Things, considering the cloud computing paradigm.



CAPÍTULO 7: BIBLIOGRAFÍA

- [1] Python, Página Oficial, Disponible on-line: [Python.org](https://python.org), (accedido Junio 2016).
- [2] Numpy, Wikipedia. Disponible on-line: es.wikipedia.org, (accedido Junio, 2016).
- [3] Tesseract OCR, Disponible on-line: es.wikipedia.org, (accedido Junio 2016).
- [4] F. Sancho, Aprendizaje automático, Disponible on-line: cs.us.es/~fsancho/, (Accedido Junio 2016).
- [5] OpenCv, OpenCV, Disponible on-line: opencv.org, (accedido Junio 2016).
- [6] G.Pajares, J.M Cruz, Visión por Computador: imágenes digitales y aplicaciones. RA-MA, Madrid (2007).
- [7] G. Pajares y J.M. Cruz (Eds), Aprendizaje Automático, RA-MA, Madrid, (2010).
- [8] Madrileña Red de Gas. Disponible on-line: Madrilena.es, (accedido Junio 2016).
- [9] The Matworks, MATLAB, Librerías MATLAB, Disponible on-line: <http://es.mathworks.com/products/matlab/>, (accedido Junio 2016).
- [10] Adrian Rosebrock, OpenCV, Disponible on-line: pyimagesearch.com, (accedido Junio 2016).
- [11] OpenCV, tutorial OpenCV, Disponible on-line: opencv-python-tutorials.readthedocs.io, (accedido Junio 2016).
- [12] Patrón Modelo-Vista-Controlador Disponible on-line: es.wikipedia.org, (accedido Junio 2016).
- [13] MongoDB. Tutorial mongoDB, Disponible on-line: tutorialspoint.com, (accedido Junio 2016).
- [14] Wikipedia. Procesamiento de imágenes, Disponible on-line: es.wikipedia.org, (accedido Junio 2016).



- [15] PhotoImage, Disponible on-line: <http://effbot.org/tkinterbook/photoimage.html>, (accedido Junio 2016).
- [16] MongoDB. Tutorial mongoDB, Disponible on-line: api.mongodb.com (accedido Junio 2016).
- [17] Método basado en Umbralización, Disponible on-line: tratamiento-de-imagenes.blogspot.com.es, (accedido Junio 2016).
- [18] Wikipedia, Disponible on-line: umbralización, es.wikipedia.org, (accedido Junio 2016).
- [19] Pavon, J. Patron Modelo Vista Controlador, Disponible on-line: fdi.ucm.es/profesor/jpavon/poo/2.14.MVC.pdf (acceso Junio 2016).
- [20] Otsu The Mathworks, Disponible on-line: mathworks.com, (accedido Junio 2016).
- [21] zetCode tkinter tutorial, Disponible on-line: zetcode.com/gui/tkinter/ (accedido junio 2016).
- [22] Django, tkinter, Disponible on-line: effbot.org/tkinterbook/ (accedido Junio 2016).
- [23] I. Hidalgo, R. J. Sánchez. Reconocimiento de caracteres mediante imágenes en contadores de gas en entornos reales. Proyecto fin de Grado, Facultad de Informática. Universidad Complutense de Madrid (2015).
- [24] Sentdex, tkinter. Disponible on-line: youtube.com, (accedido Junio 2016).
- [25] JetBrains. Disponible on-line: Jetbrains.com, (accedido Junio 2016).
- [26] G. Pajares, Percepción computacional. Apuntes de la asignatura 2016).
- [27] Wikipedia, reconocimiento óptico de caracteres, Disponible on-line: es.wikipedia.org, (2016).
- [28] Método del Valor Umbral, Disponible on-line: es.wikipedia.org, (accedido Junio 2016).
- [29] Universidad Nacional de Quilmes, Apertura Morfológica, Disponible on-line: <http://iaci.unq.edu.ar/>, (accedido Junio 2016).



- [30] González, G. Transformada de Fourier, Disponible on-line
<https://www.emis.de/journals/DM/v5/art6.pdf>, (accedido Junio 2016).
- [31] Wikipedia, filtrado en el dominio espacial. Disponible on-line:
<Es.wikipedia.org> (accedido junio 2016).
- [32] CakePhp MVC, Disponible on-line:
<http://book.cakephp.org/2.0/es/cakephp-overview/understanding-model-view-controller.html>, (accedido Junio 2016).
- [33] JSON, JSON, Disponible on-line: www.json.com, (accedido Junio 2016).
- [34] NoSQL, NoSQL, Disponible on-line: nosql-database.org , (accedido Junio 2016).
- [35] BSON,BSON, Disponible on-line: bsonspec.org, (accedido Junio 2016).
- [36] D. Robin, Github. Disponible on-line: github.com/RobinDavid, (accedido Junio 2016).





CAPÍTULO 8: ANEXO

A) Manual de Instalación

A continuación se explica la instalación del software que se entregó junto con la memoria para la correcta puesta en funcionamiento del proyecto en el sistema operativo Windows al tutor.

El primer paso que se ha de efectuar es descomprimir el archivo adjunto en el DVD.

- **Contenido Archivo Adjunto**

- Ejecutable de Python 2.7.11.
- Ejecutable de MongoDB 3.0.7.
- Ejecutable Tesseract 3.02.02.
- Librería OpenCV 2.4.11.
- Librería numpy 1.7.1.

- **Instalación del Software Adjunto**

Instalación Python 2.7

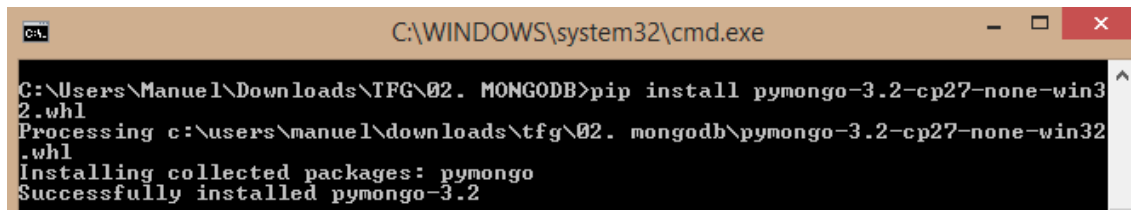
Para instalar el compilador del lenguaje de programación que se ha utilizado sólo hay ejecutar el instalador que incluye el DVD y en las opciones de instalación debemos habilitar la opción “ADD PATH”.

Instalación de mongoDB 3.07 versión completa

Para la instalación de la Base de Datos en el escritorio donde se va a trabajar se debe ejecutar el instalador de MongoDB disponible en el software adjunto los pasos serán los siguientes:

- Ejecutamos el instalador de MONGODB.
- Instalamos Pymongo utilizando la herramienta pip.

En la figura 23 se puede apreciar la consola de ejecución.



```
C:\WINDOWS\system32\cmd.exe
C:\Users\Manuel\Downloads\TFG\02. MONGODB>pip install pymongo-3.2-cp27-none-win32.whl
Processing c:\users\manuel\downloads\tfg\02. mongodb\pymongo-3.2-cp27-none-win32.whl
Installing collected packages: pymongo
Successfully installed pymongo-3.2
```

Figura 23: Consola de ejecución instalación MongoDB.

- Abrir una ventana de Prompt con el comando “cmd” y ejecutar el siguiente comando en la ventana para que instalar el paquete necesario.
- El siguiente paso es crear una carpeta para el servidor MONGO desde el mismo cmd que ya tenemos abierto, para ello se utiliza el siguiente comando:

`C:\mongodb\bin\mongod.exe --dbpath RUTA DE LA CARPETA`

- Ejecutar mongo.exe.

Se ha creado un .bat que ejecuta directamente mongod.exe y mongo.exe como si fuese un ejecutable, se puede editar el .bat para escribir la ruta que vamos a utilizar.

Si al intentar abrir mongod y mongo.exe la ventana se cierra al momento seguramente se cerró mal una conexión anterior y habrá que borrar en la carpeta data el archivo **mongod.lock**.

Instalación librería Opencv 2.4.11

Para la instalación de la librería de Python usado para el análisis de las imágenes se debe:

- Descomprimir el archivo “**open cv 2.4.11**” en el directorio raíz **C:**
- Copiar el contenido de **C:\opencv\build\python\x86\2.7** en **C:\Python27\Lib\site-packages**

Instalamos la librería NumPy 1.7.1

Para la instalación de la librería de Python que se va a usar para dar mayor soporte a las matrices y vectores de Python los pasos a seguir son:

- Ejecutar el instalador de NumPy adjunto compatible con la versión de Python ya instalada.



Instalación Tesseract OCR

Para la instalación de Tesseract OCR, software necesario para el reconocimiento óptico de caracteres, necesitamos:

- Ejecutar el instalador de Tesseract ocr 3.02.02 que está incluido en el DVD adjunto.

B) Despliegue de la aplicación

- El primer paso del despliegue de la aplicación en el ordenador es abrir el ejecutable de MongoDB para lanzar la base de datos. Se puede especificar la ruta de almacenamiento modificando en el archivo el siguiente comando:

C:\mongodb\bin\mongod.exe --dbpath RUTA DE NUESTRA CARPETA

- Una vez iniciada la base de datos solo hace falta lanzar la aplicación ya sea a través del ejecutable o desde un compilador Python para comenzar a utilizarla.

C) Manual de Uso aplicación de escritorio

En esta sección se detalla el uso de la aplicación paso a paso, de manera que el usuario no tenga ninguna duda de cómo utilizar cualquiera de las funcionalidades disponibles.

Incluye una descripción del menú principal, el procesamiento de imágenes a través de la carga, recorte y tratamiento de la imagen, almacenamiento en base de datos, histórico de cliente y listado de todos los clientes.

Menú principal: pantalla donde el usuario de la aplicación puede seleccionar la opción que desea ejecutar. En la figura 24 se aprecia el contenido del menú de la aplicación.



Figura 24: Captura de pantalla del menú principal de la aplicación.

Las opciones que dispone el menú están numeradas de la siguiente manera:



Home:

Si se selecciona este icono se volverá a este menú principal.



Procesar imagen:

Al seleccionar esta opción se accede al área de carga imagen para leer una medición de contador y cliente.



Histórico de cliente:

Este icono muestra la pantalla donde introduciendo un número de usuario muestra sus mediciones.



Listar usuarios:

Este icono muestra el listado con todos los clientes de la empresa.



Salir:

Este icono será el encargado de cerrar la aplicación.

A continuación, si se ha seleccionado alguna de las opciones referentes a los casos de uso descritos en el apartado 4, los pasos a seguir son los siguientes:

- **Tratamiento de la imagen**

En esta pantalla se muestran los distintos botones con las funcionalidades. En la Figura 25 podemos ver cómo sería la pantalla principal cuando se decide acceder desde el menú al caso de uso tratamiento de imágenes:



Figura 25: Pantalla principal para el caso de tratamiento de imágenes.

A continuación se desglosan las funciones asociadas a cada uno de los botones.

Lo primero que se ha de hacer es cargar la imagen, para ello se pulsa el botón Abrir Imagen, eligiendo la ruta donde se ubica la imagen. La figura 26 representa la ventana que se despliega para elegir una imagen una vez seleccionada la opción correspondiente.

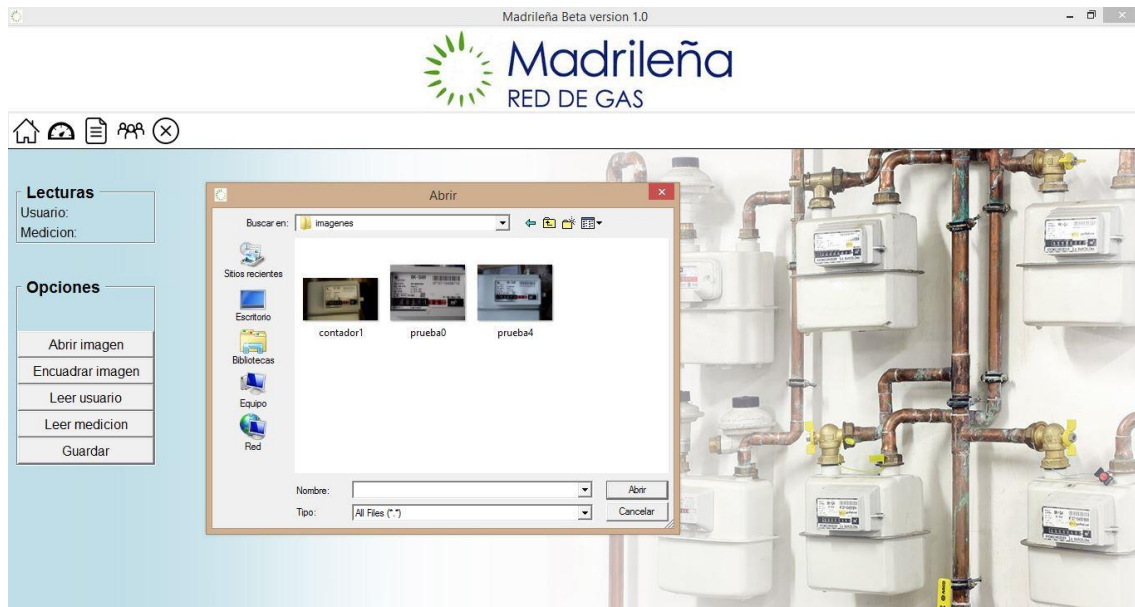


Figura 26: Seleccionar imagen para cargar en el programa.

Una vez se elige la opción de abrir dentro de la nueva ventana emergente donde se selecciona la imagen, ésta se carga en el programa visualizándose como se muestra en la figura 27.



Figura 27: Carga de la imagen en el programa.

Una vez cargada, se puede encuadrar el área que se desee, arrastrando el ratón por la zona deseada siempre de arriba a abajo y de izquierda a derecha.

La figura 28 muestra cómo aparece un recuadro sobre el área seleccionada.



Figura 28: Selección del área de recorte.

Una vez seleccionada el área deseada, pulsando la tecla C se realiza el corte apareciendo el área recortada en la ventana de la aplicación. La figura 29 muestra el encuadre efectuado del área seleccionada.

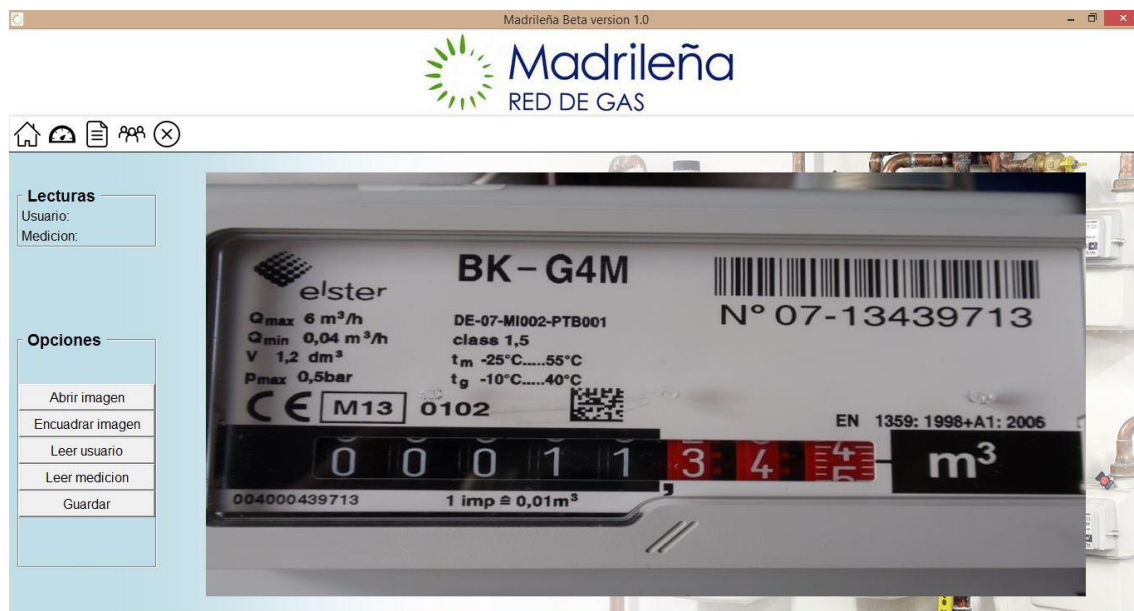


Figura 29: Encuadre de un área previamente seleccionada.

A continuación, mostrado el recuadro donde se tiene centrada el área de la lectura del contador y del número de cliente se puede proceder a leer usuario.

La figura 30 muestra la selección del área que contiene el código del cliente.

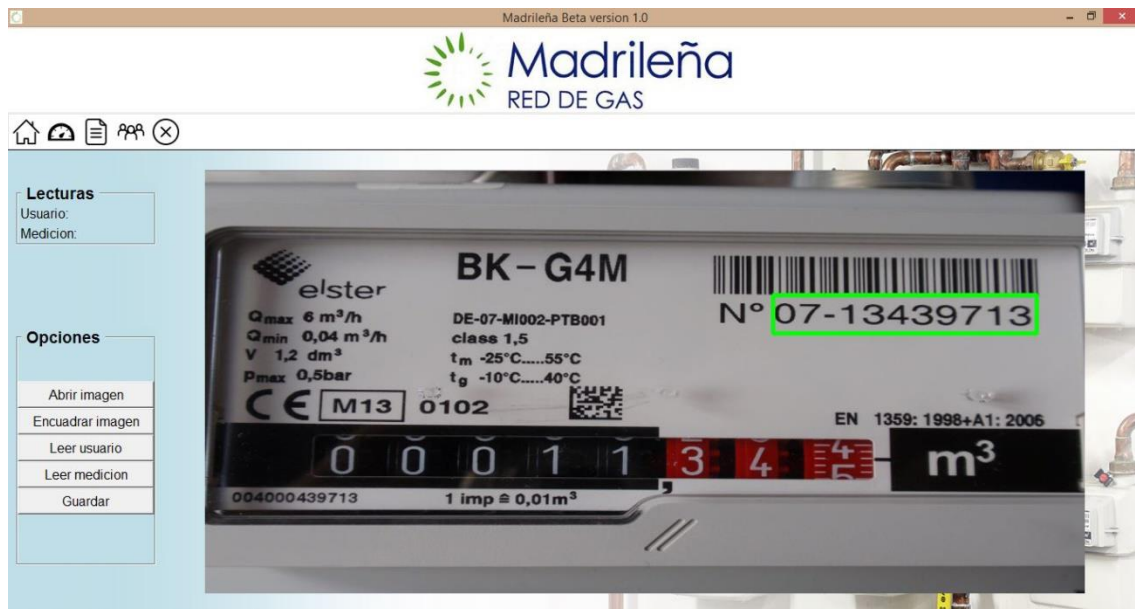


Figura 30: Selección del área donde se encuentra el código identificador del cliente.

En la figura 31 se muestra el recuadro de selección para la medición del contador, como se puede comprobar, en el apartado de lecturas, ya se han muestran los datos referentes al cliente leídos en el paso anterior.

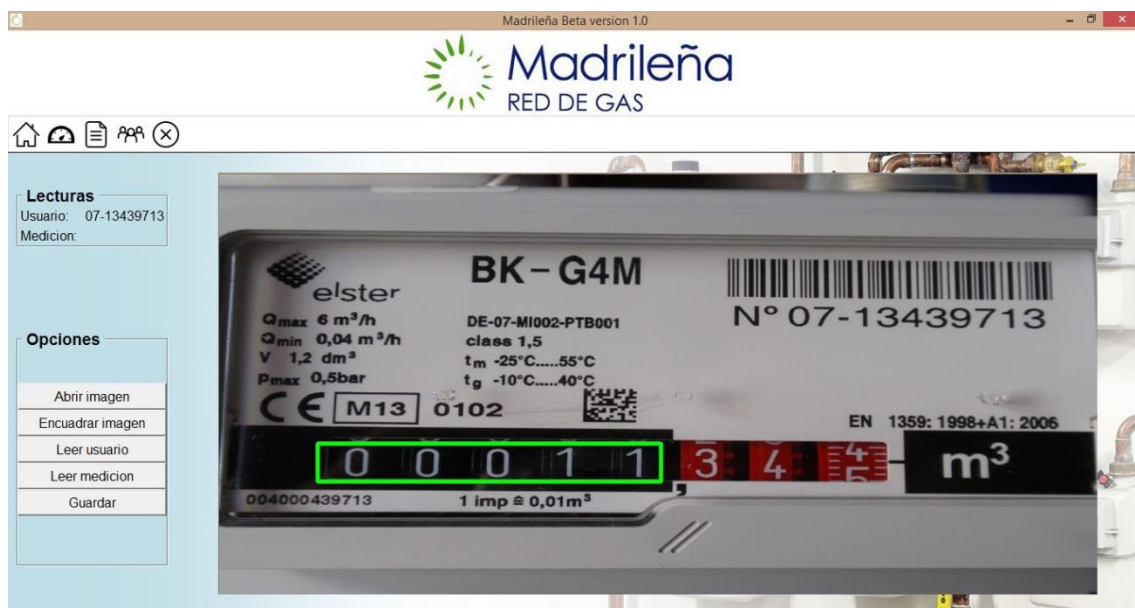


Figura 31: Selección del área donde se encuentra los datos del contador.

Si se está conforme con los datos que aparecen en el apartado de lecturas se procede al guardado la información. El programa muestra un mensaje informando de los datos que se van a almacenar, para decidir continuar con el proceso o cancelar la operación.

La figura 32 muestra, el mensaje informativo con los datos obtenidos en los dos puntos anteriores.

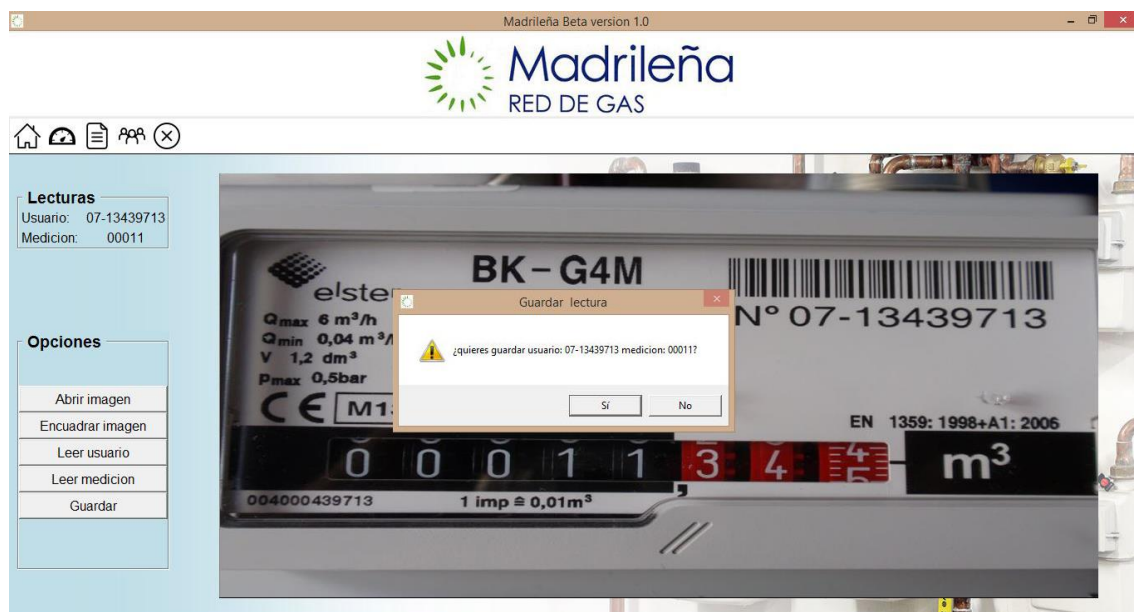


Figura 32: confirmación del guardado de los datos.

Si los datos mostrados eran correctos y el usuario ha decidido aceptar, el programa lanzará un mensaje informando si los datos se han almacenado correctamente en la Base de Datos. En caso contrario mostrará un aviso informando que los datos no han podido ser guardados con éxito.

La figura 33 muestra el mensaje de aviso de que los datos se han guardado correctamente en la base de datos.



Figura 33: Datos guardados correctamente.

- **Histórico de cliente**

Al seleccionar la opción histórico de cliente, se muestra una ventana donde se debe introducir el código de cliente que se desea listar. Si el cliente existe en la Base de Datos se muestran todas sus mediciones junto a las fechas en las que han sido procesadas.

La figura 34 muestra el histórico del cliente leído en el ejemplo anterior.



Figura 34: Ventana del histórico de un cliente de la aplicación.

- Listado de cliente

Al presionar la opción del menú de listar clientes, el usuario obtiene un listado con todos los clientes disponibles en la base de datos. En la figura 35 se puede apreciar la interfaz para el listado de los clientes contenidos en la aplicación.

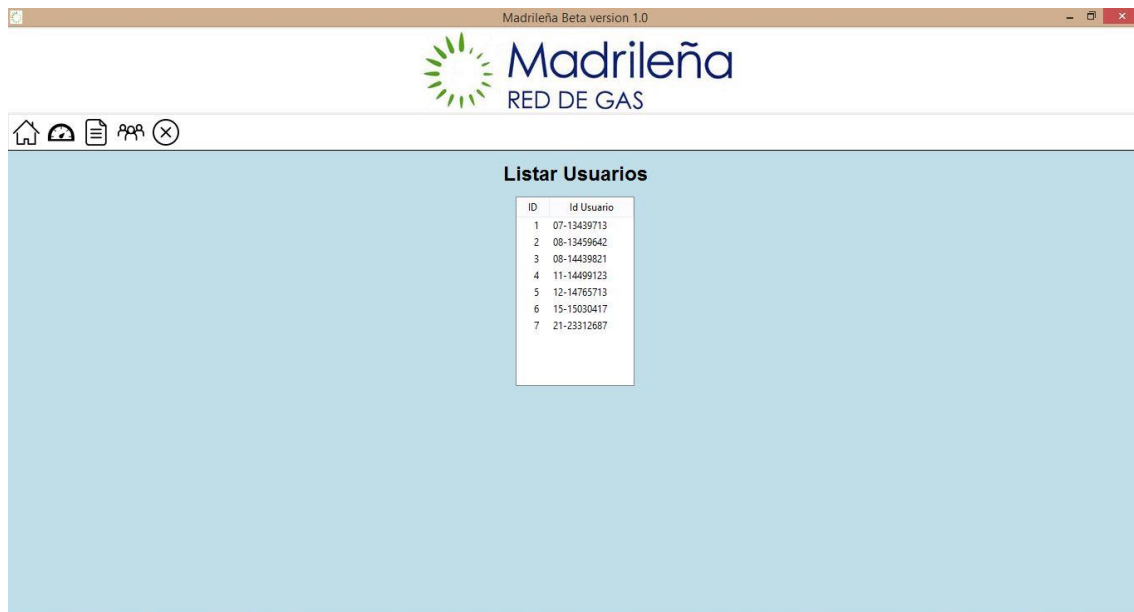


Figura 35: Ventana del listado de todos los clientes de la aplicación.

